

Introduksjon til Flux

Hva?

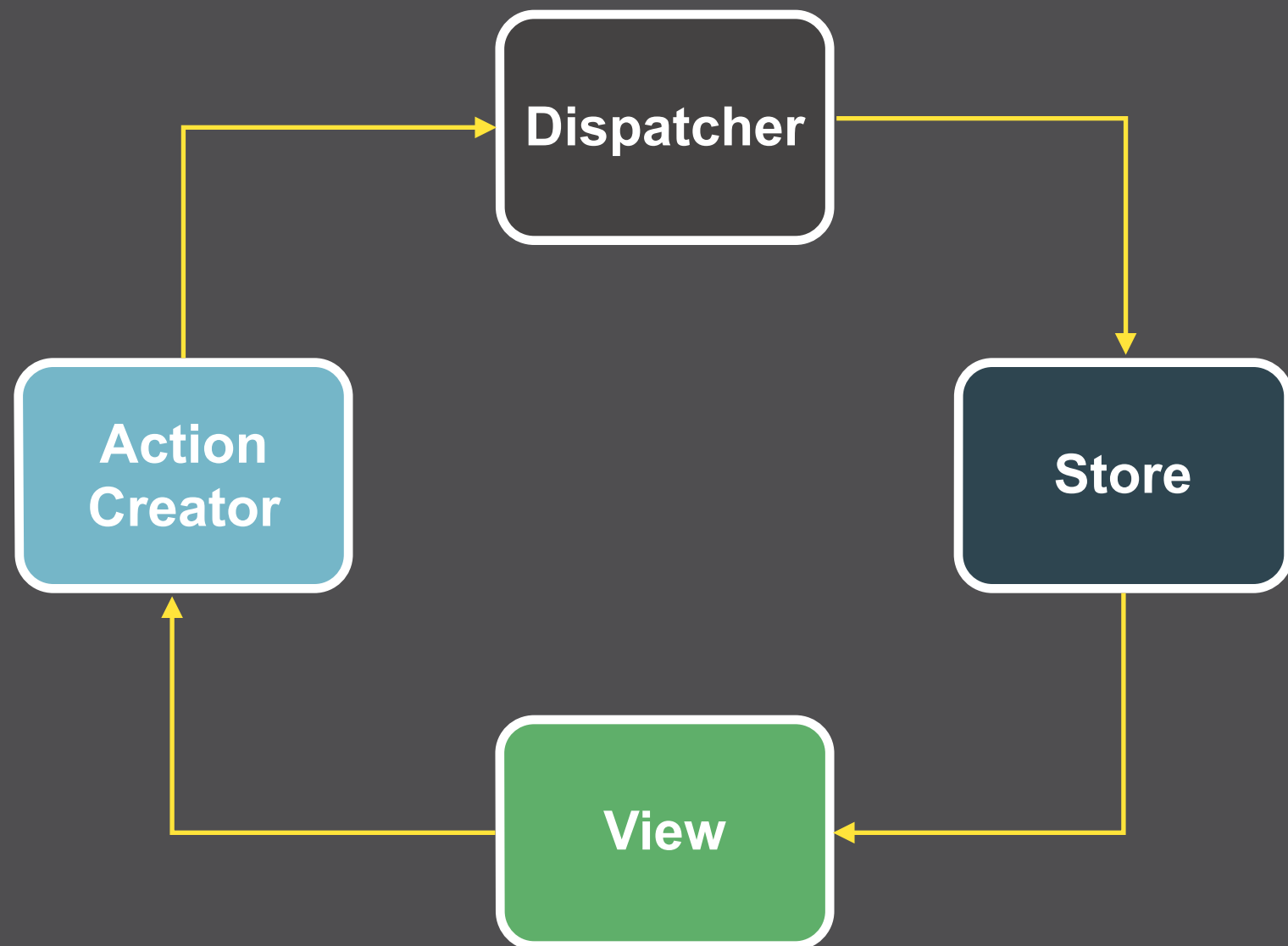
Hvordan?

Hvorfor?

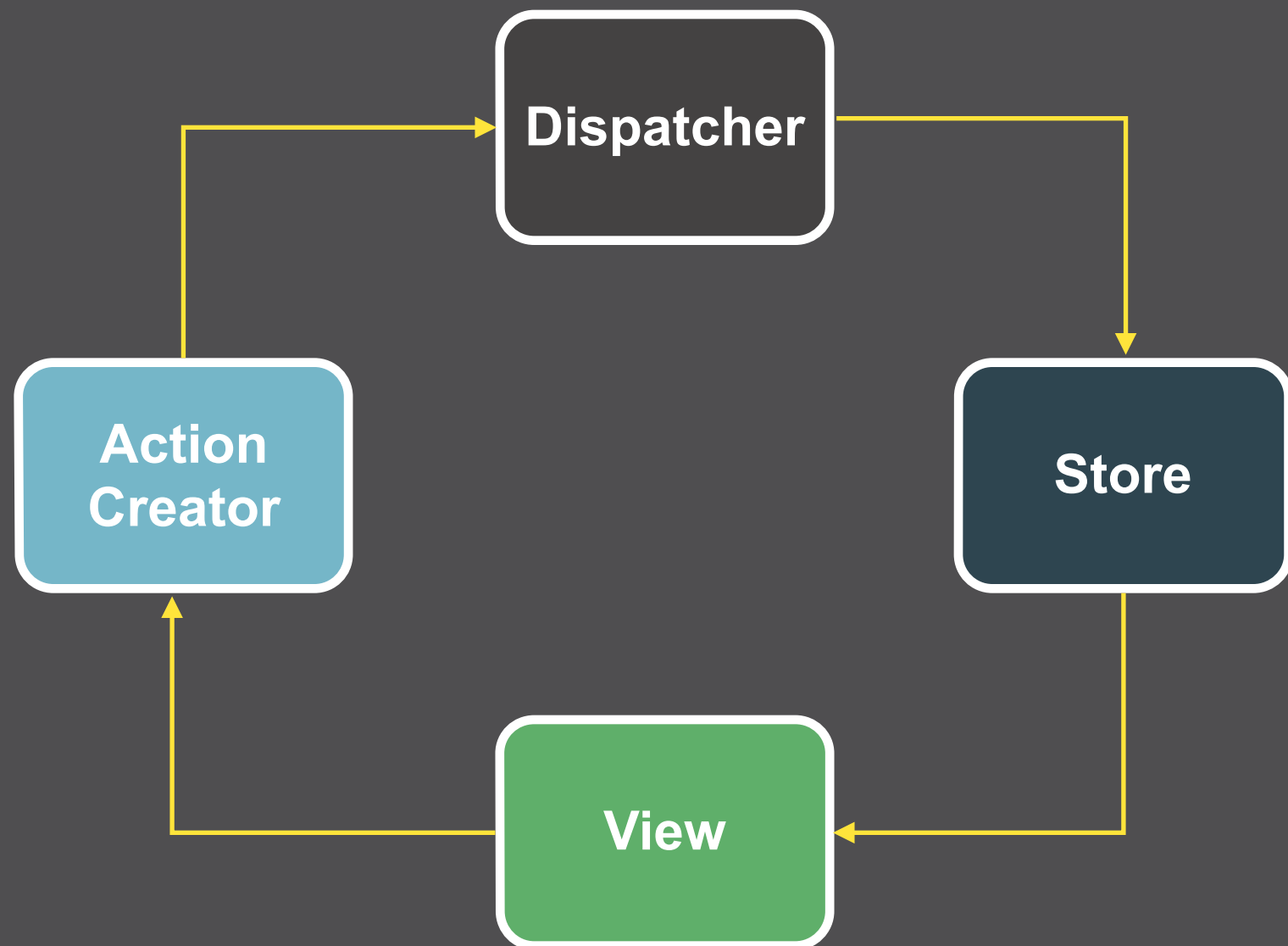
Hva?

**Et design pattern for
frontend**

Flux

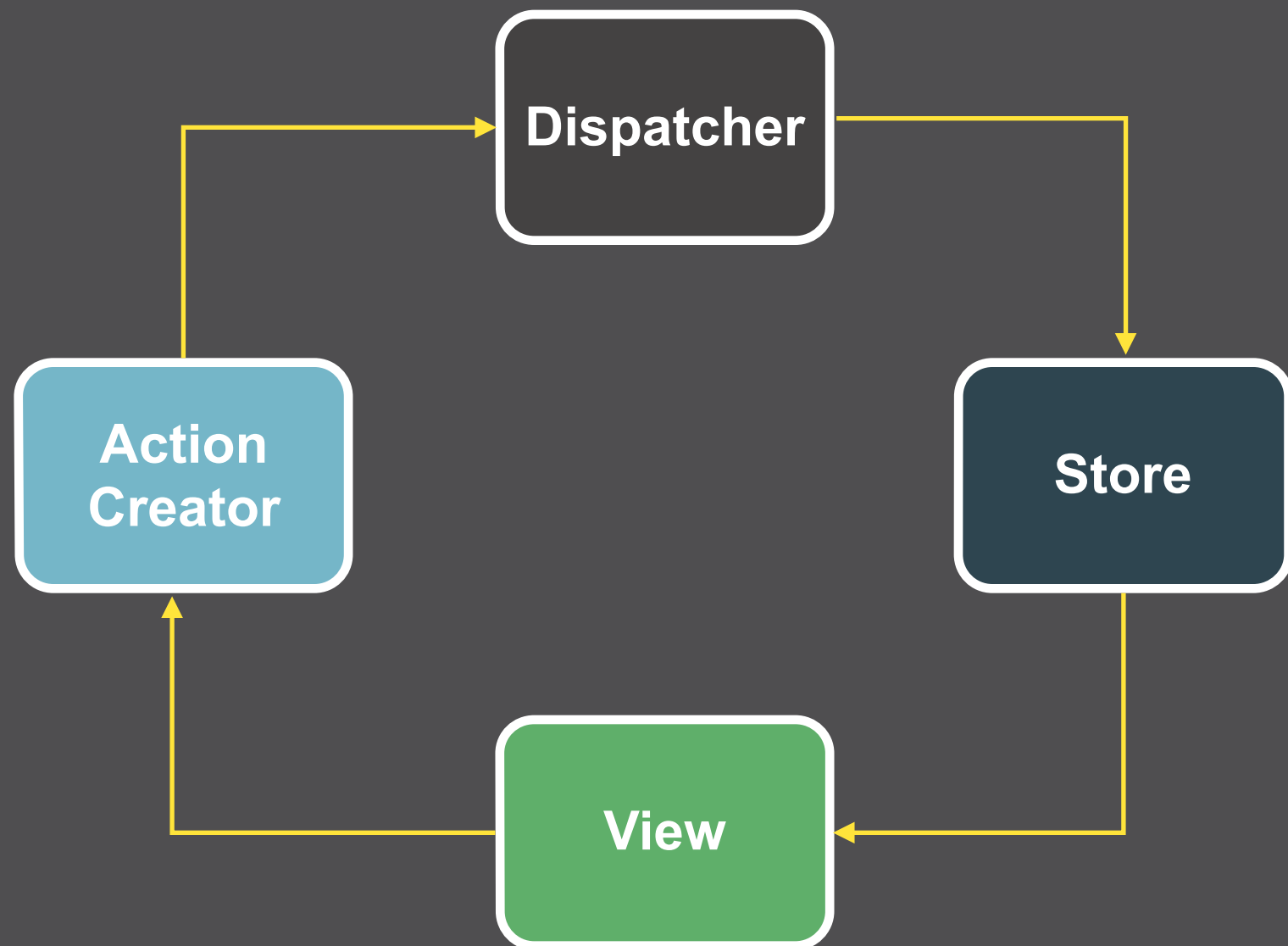


Mental modell

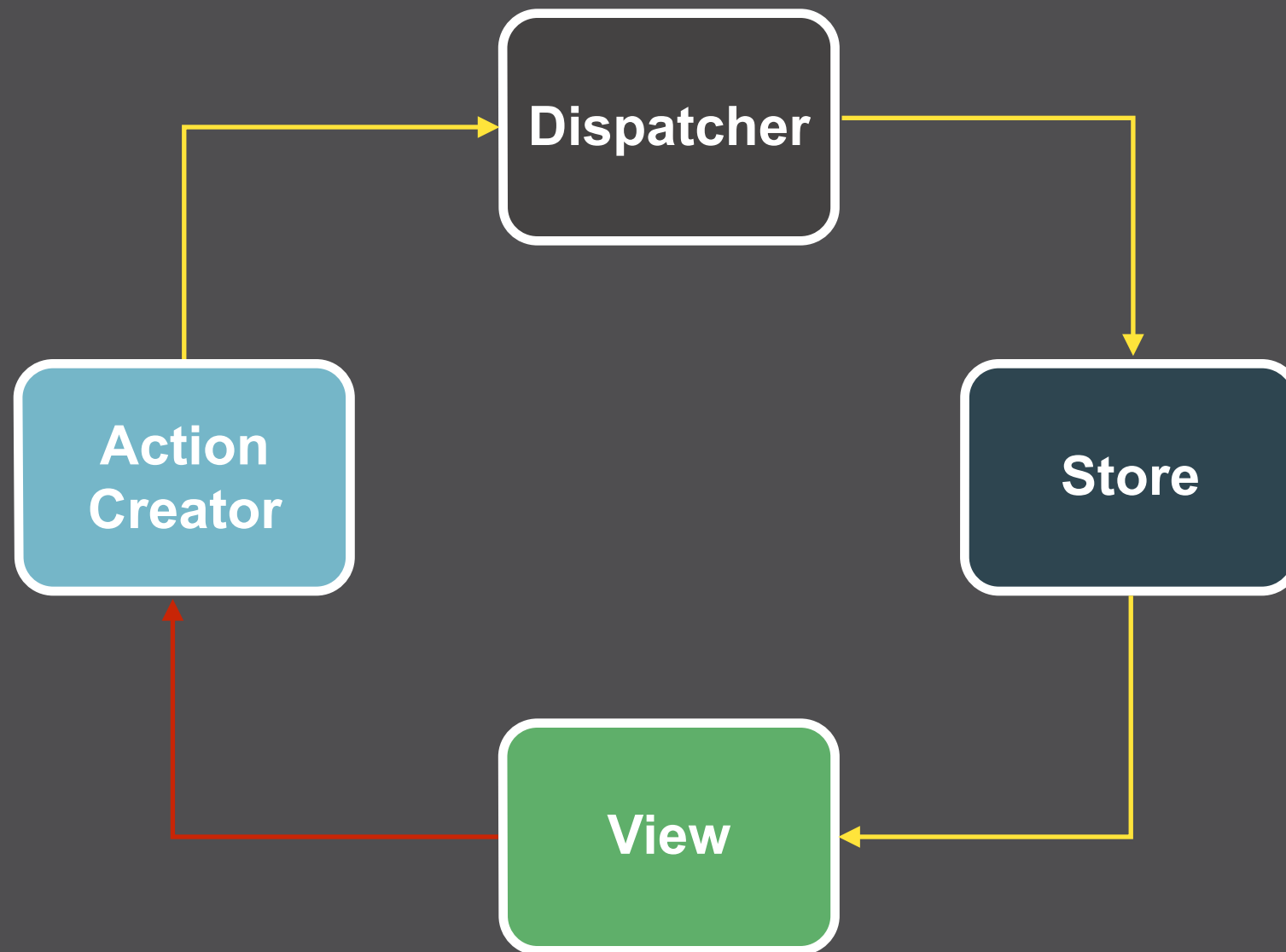


Enkelt og forutsigbart

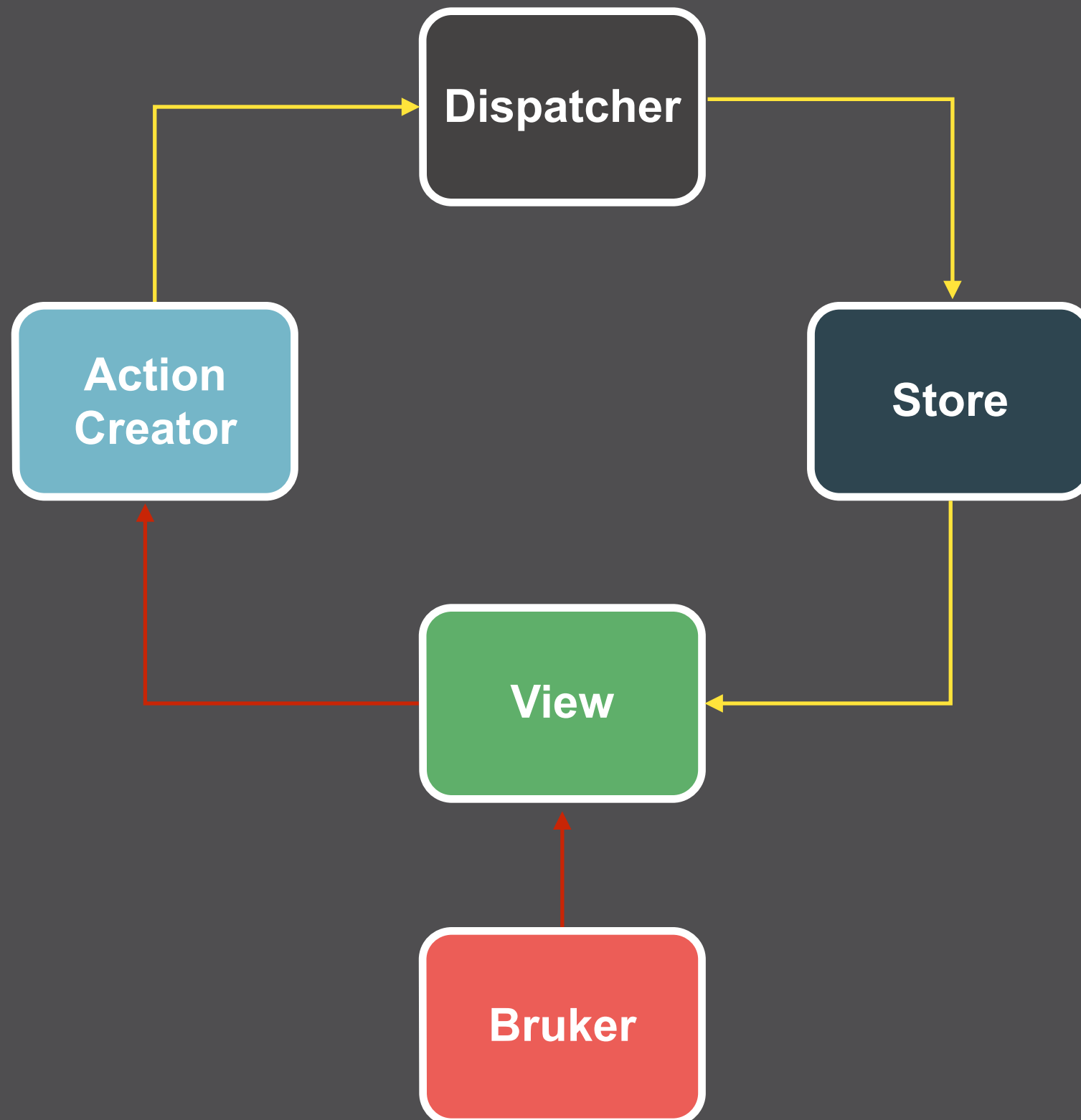
Dataflyten går kun én vei



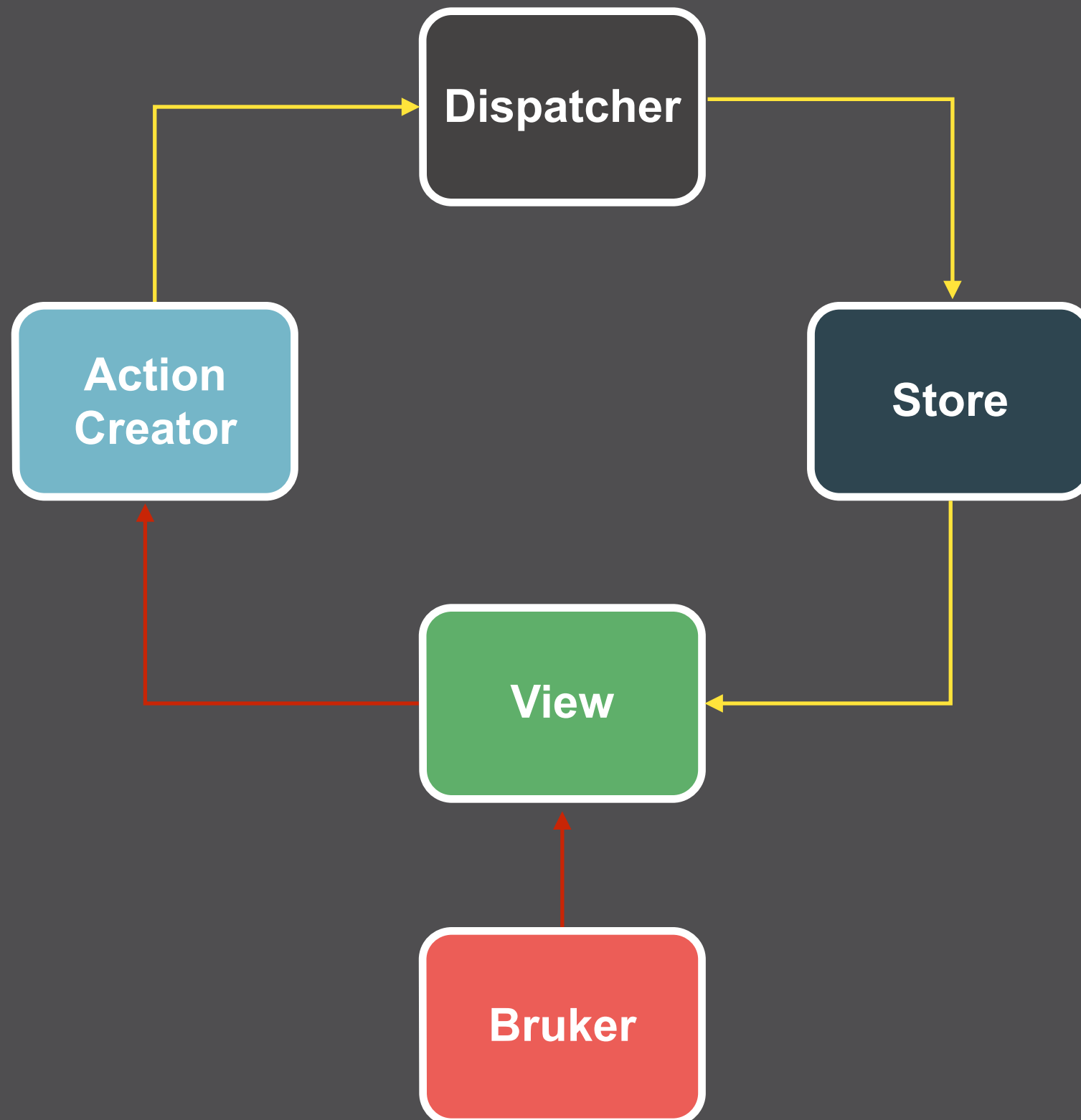
Hvor kommer data fra?

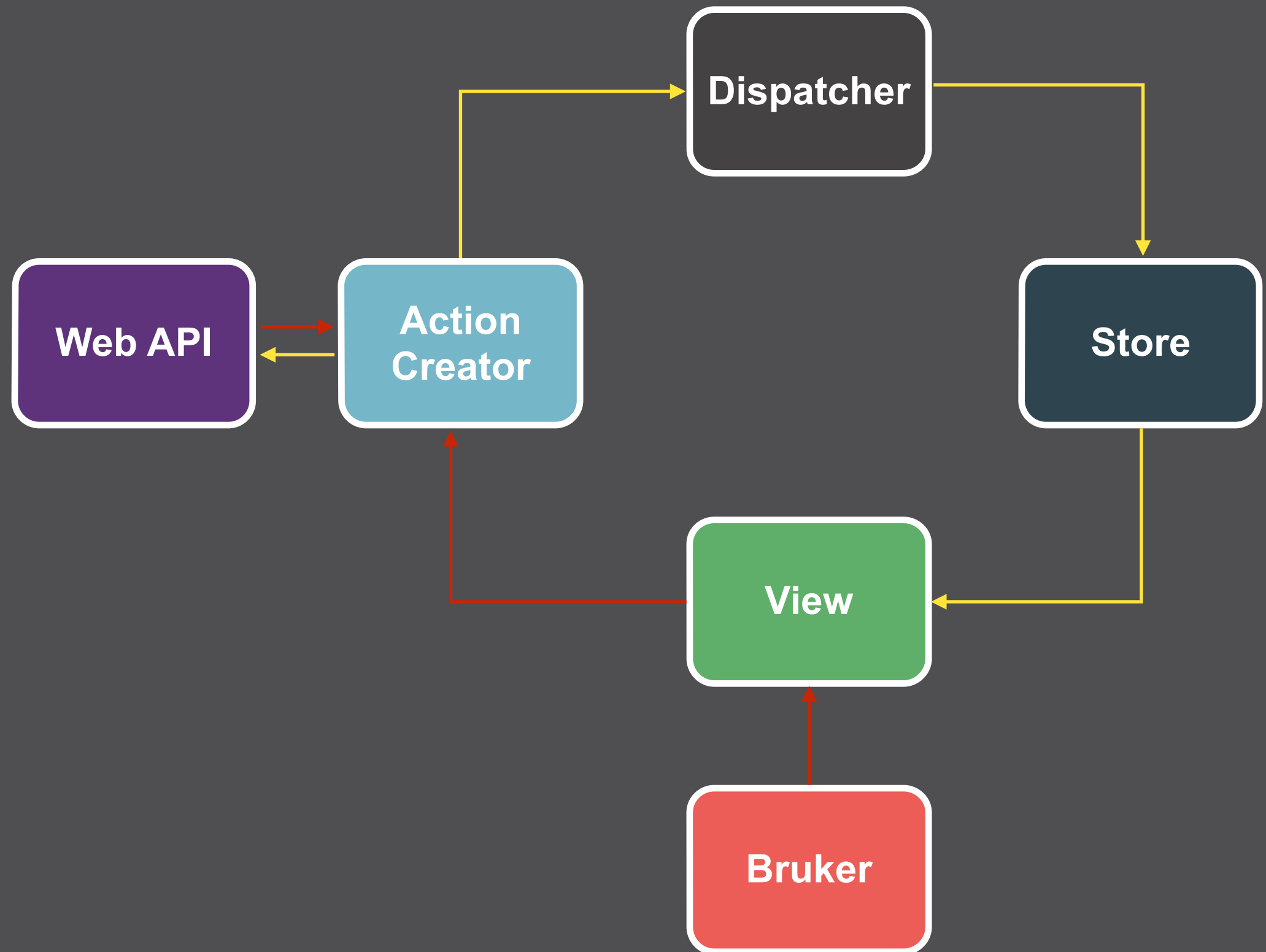


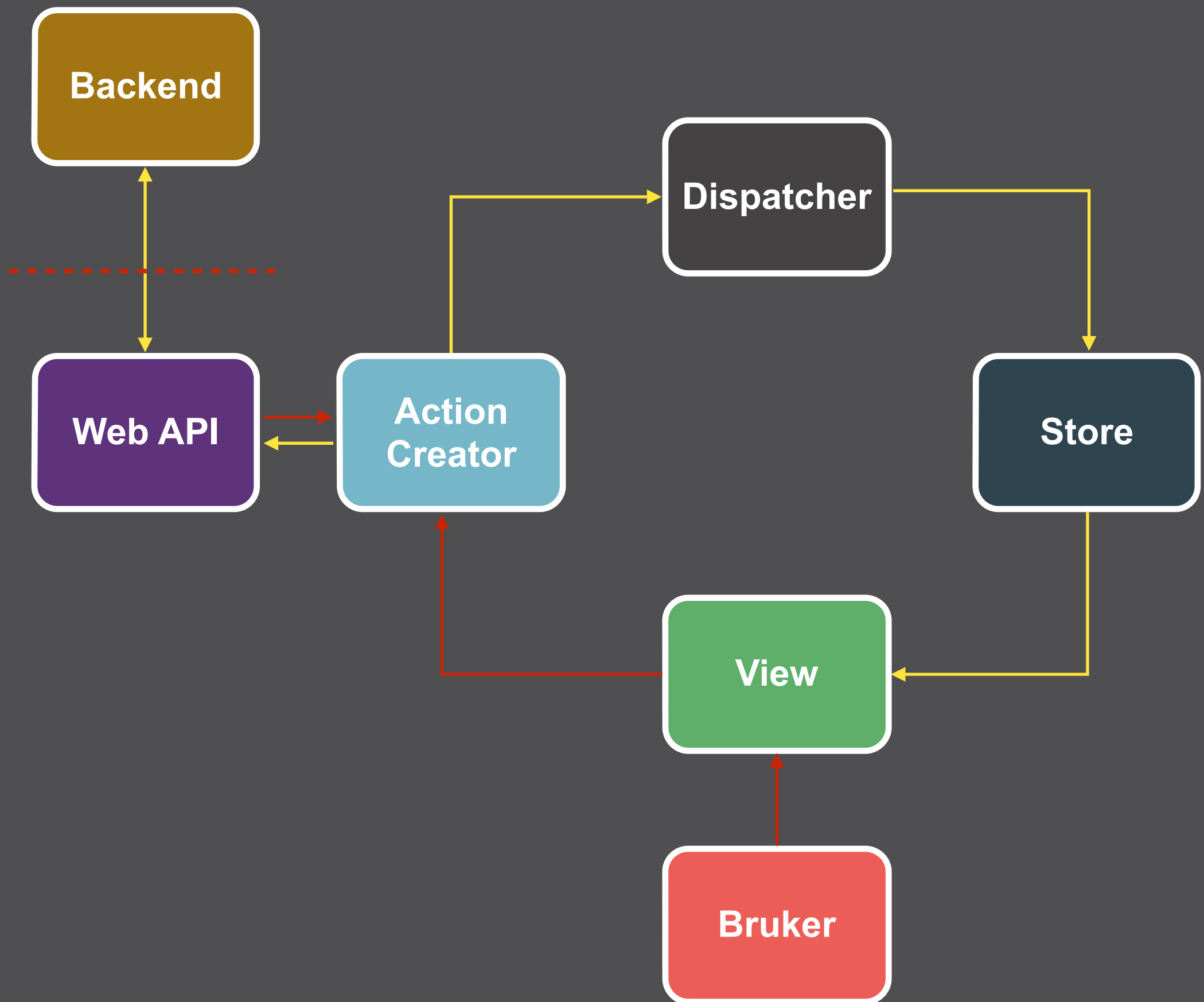
Hvor kommer data fra?

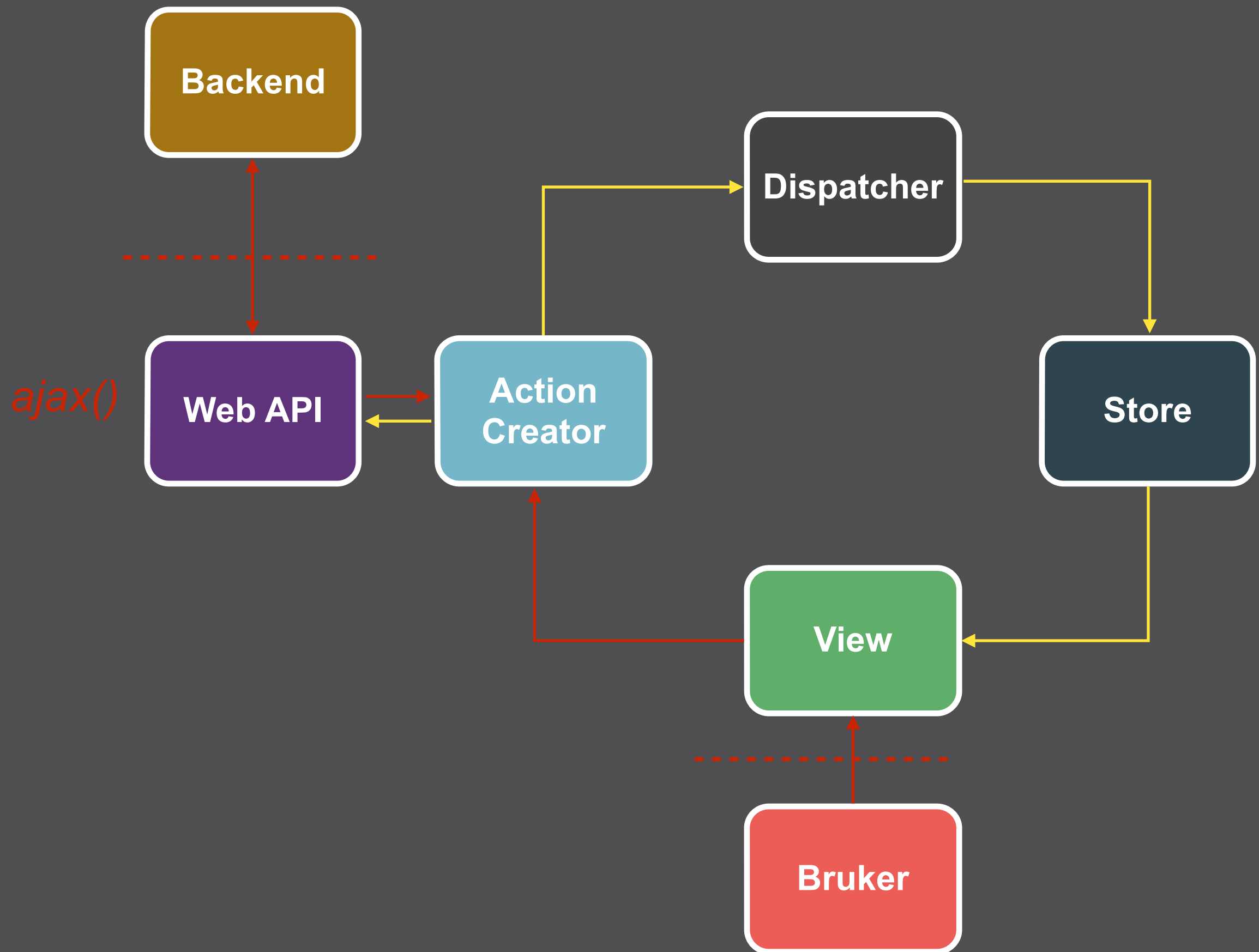


Data fra backend?

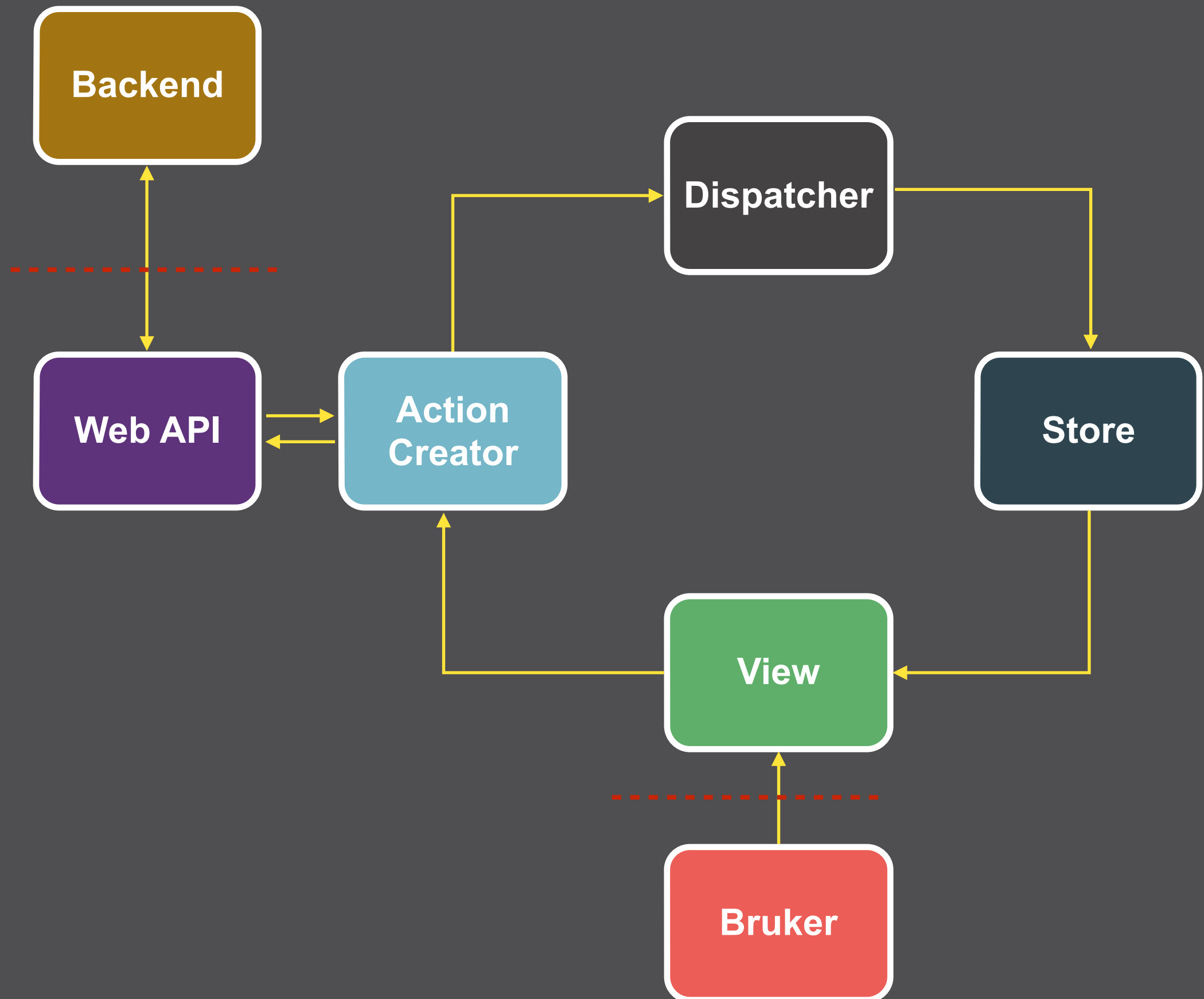




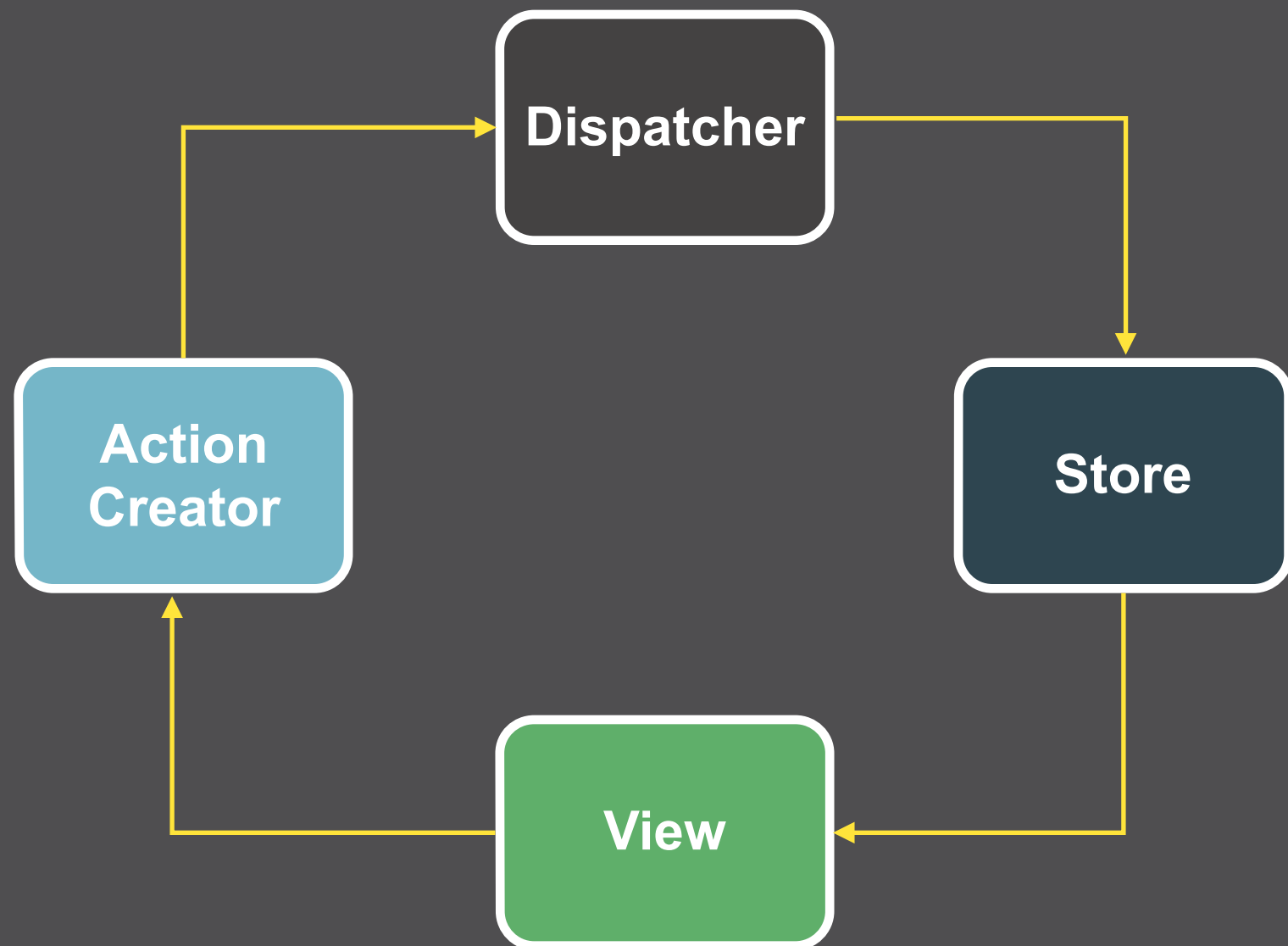




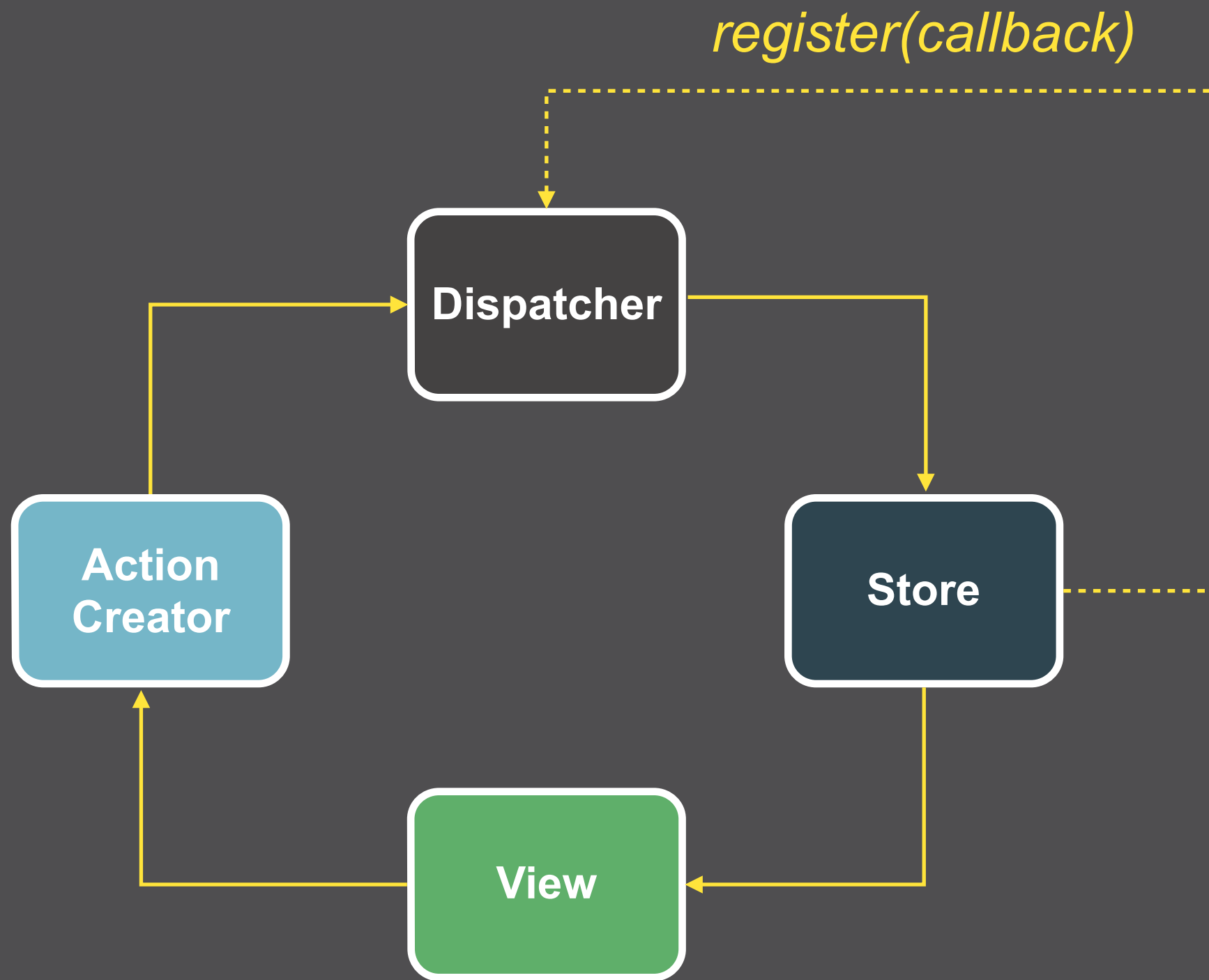
Hele arkitekturen

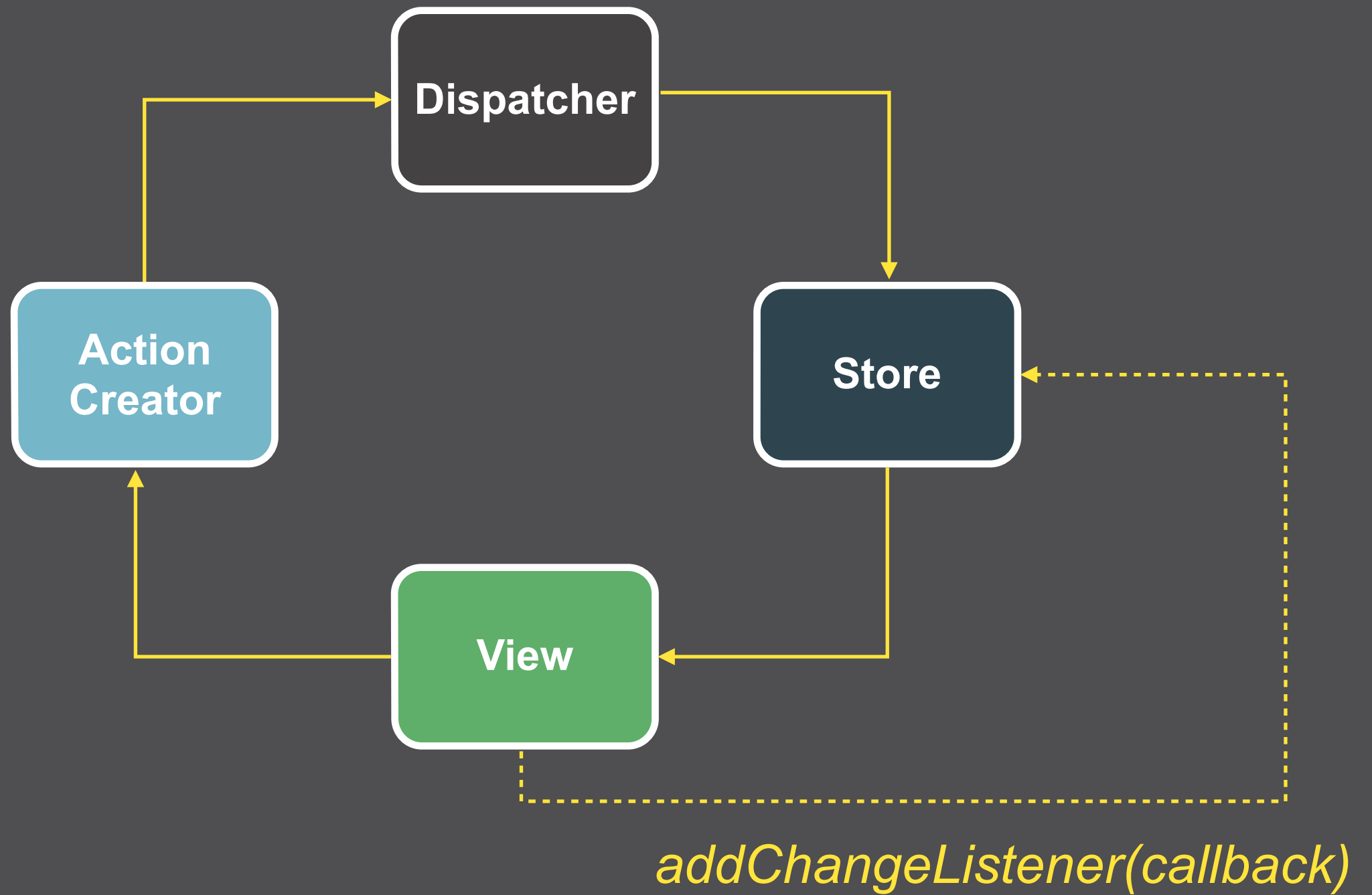


Mental modell



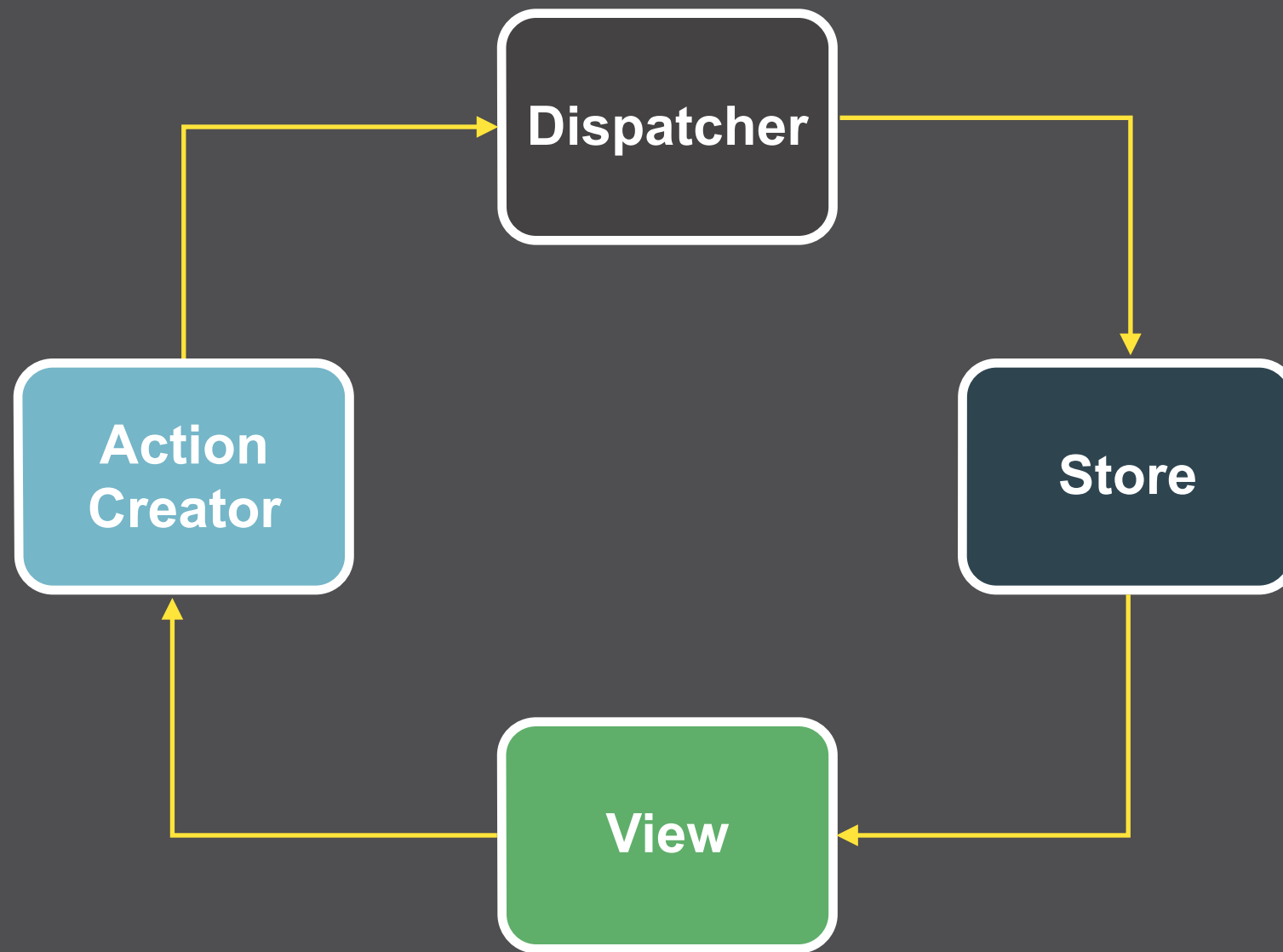
Hvordan?



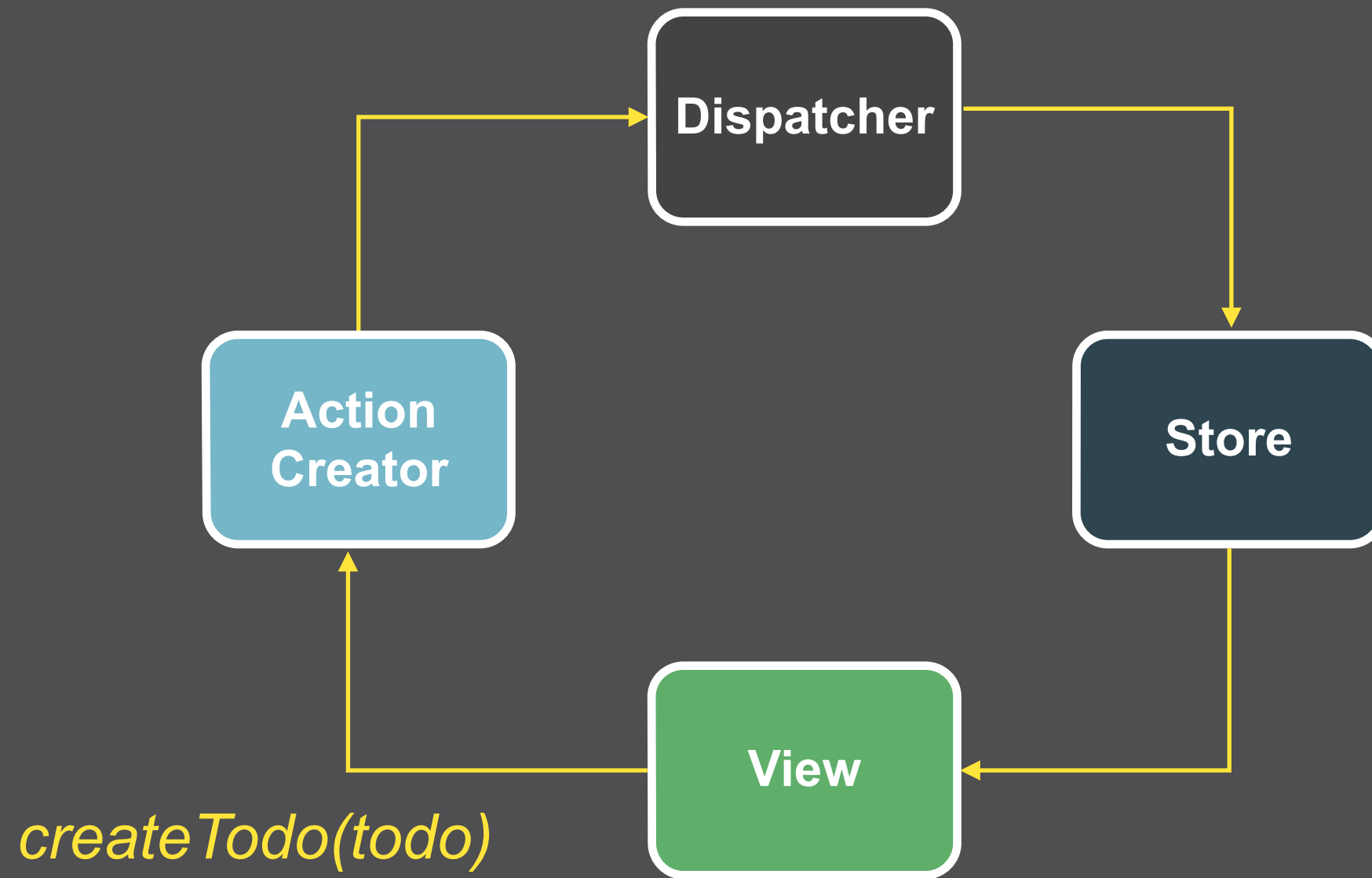


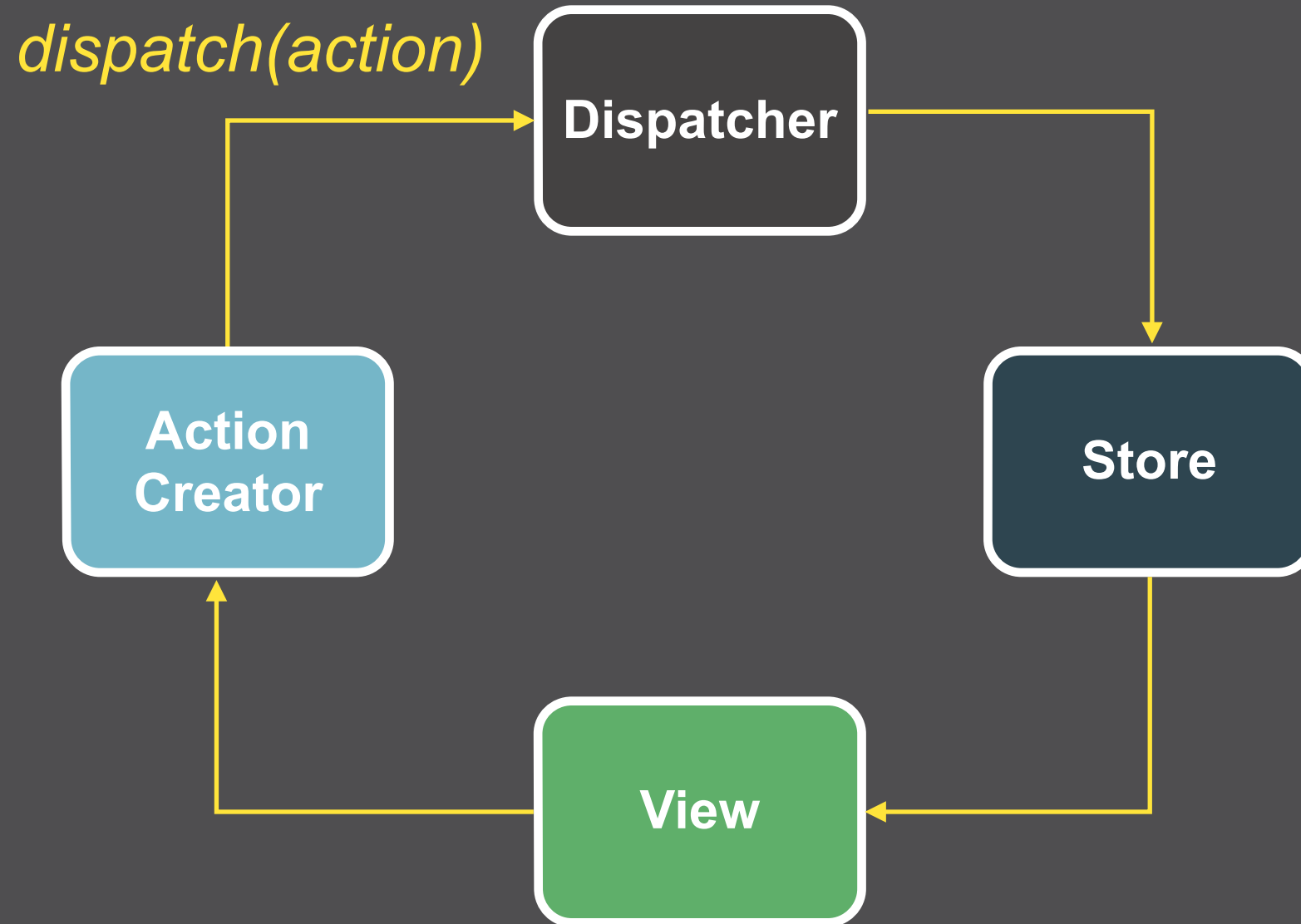
Eksempelscenario

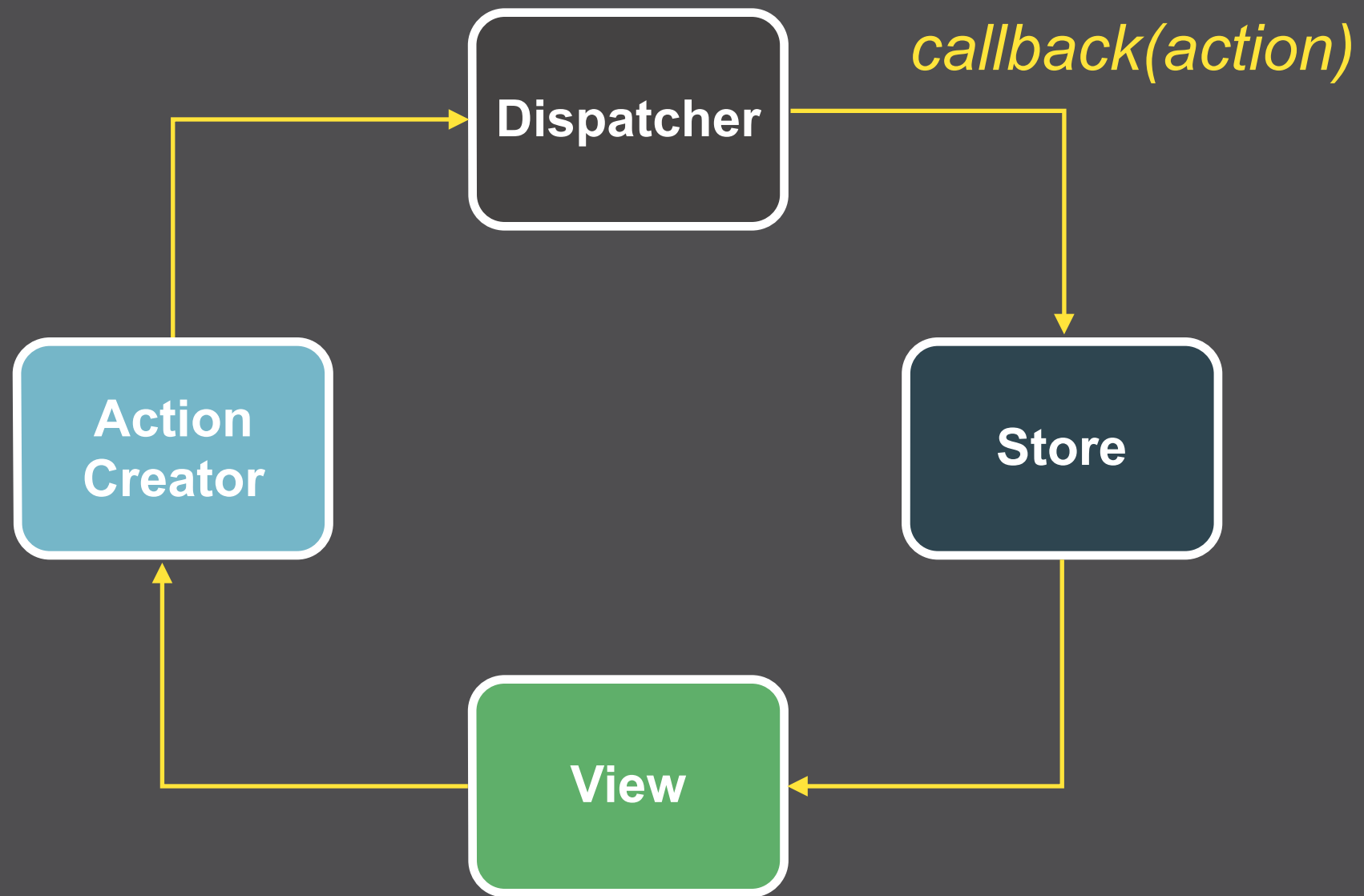
En Todo-App!

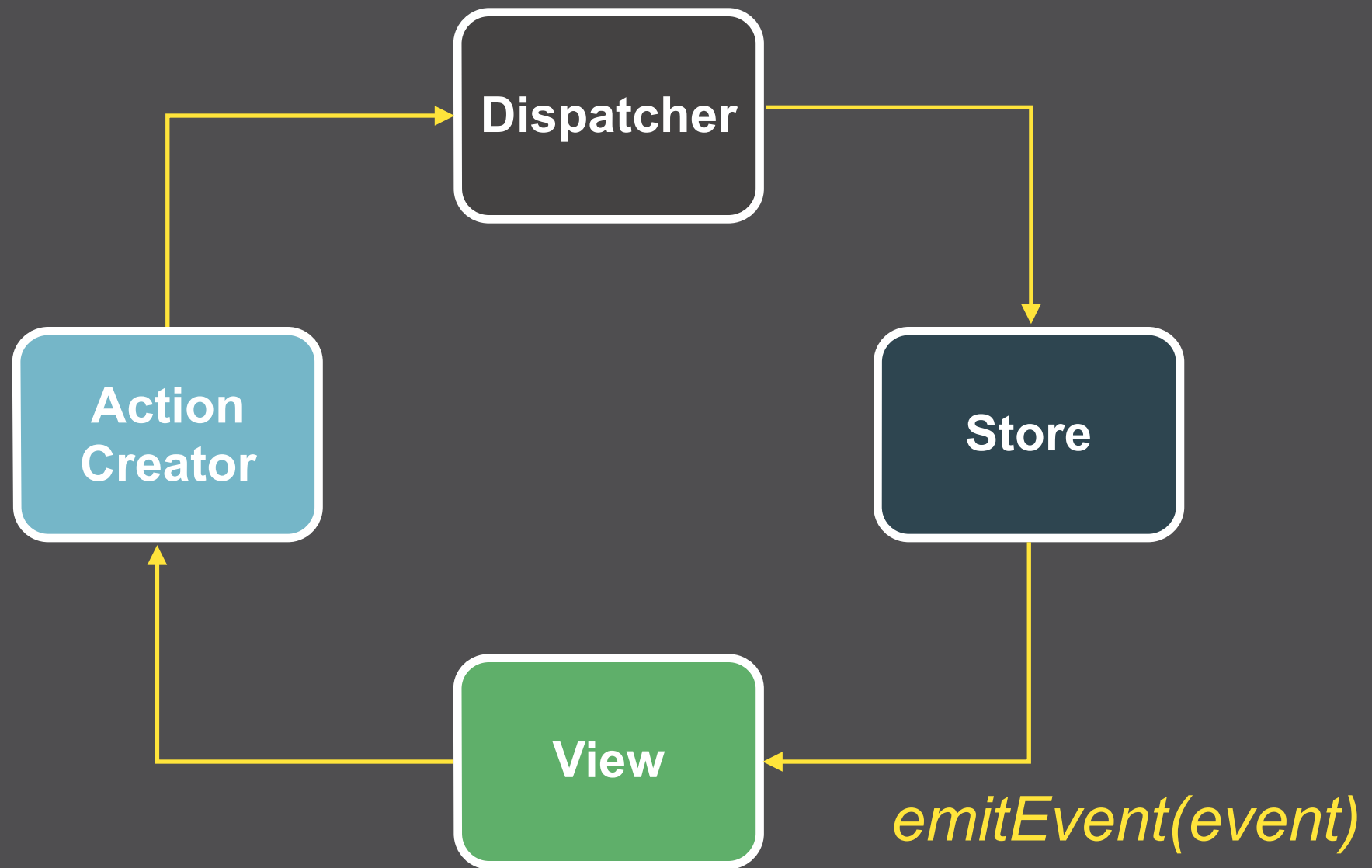


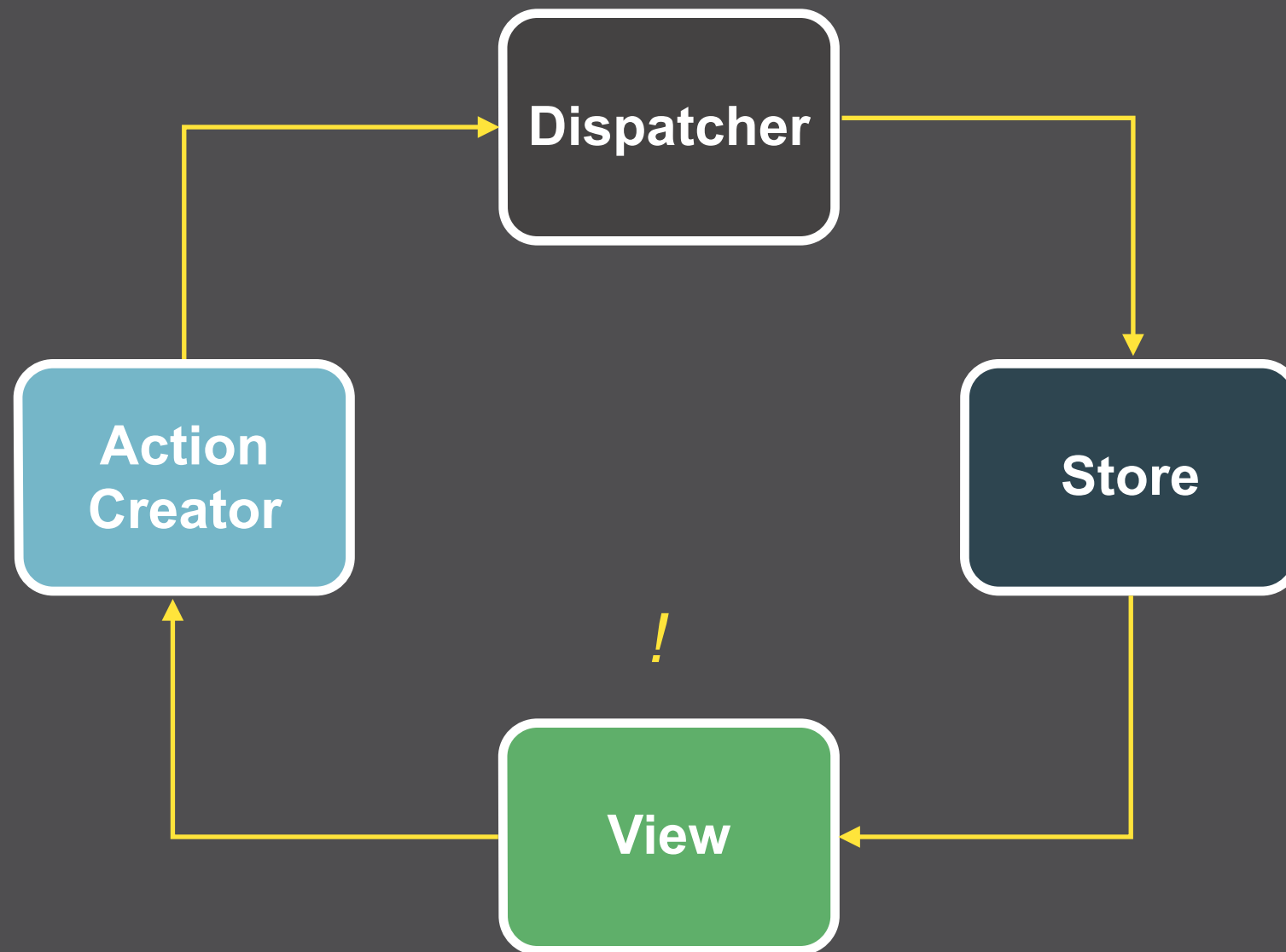
<bruker trykket på "Lag ny todo">

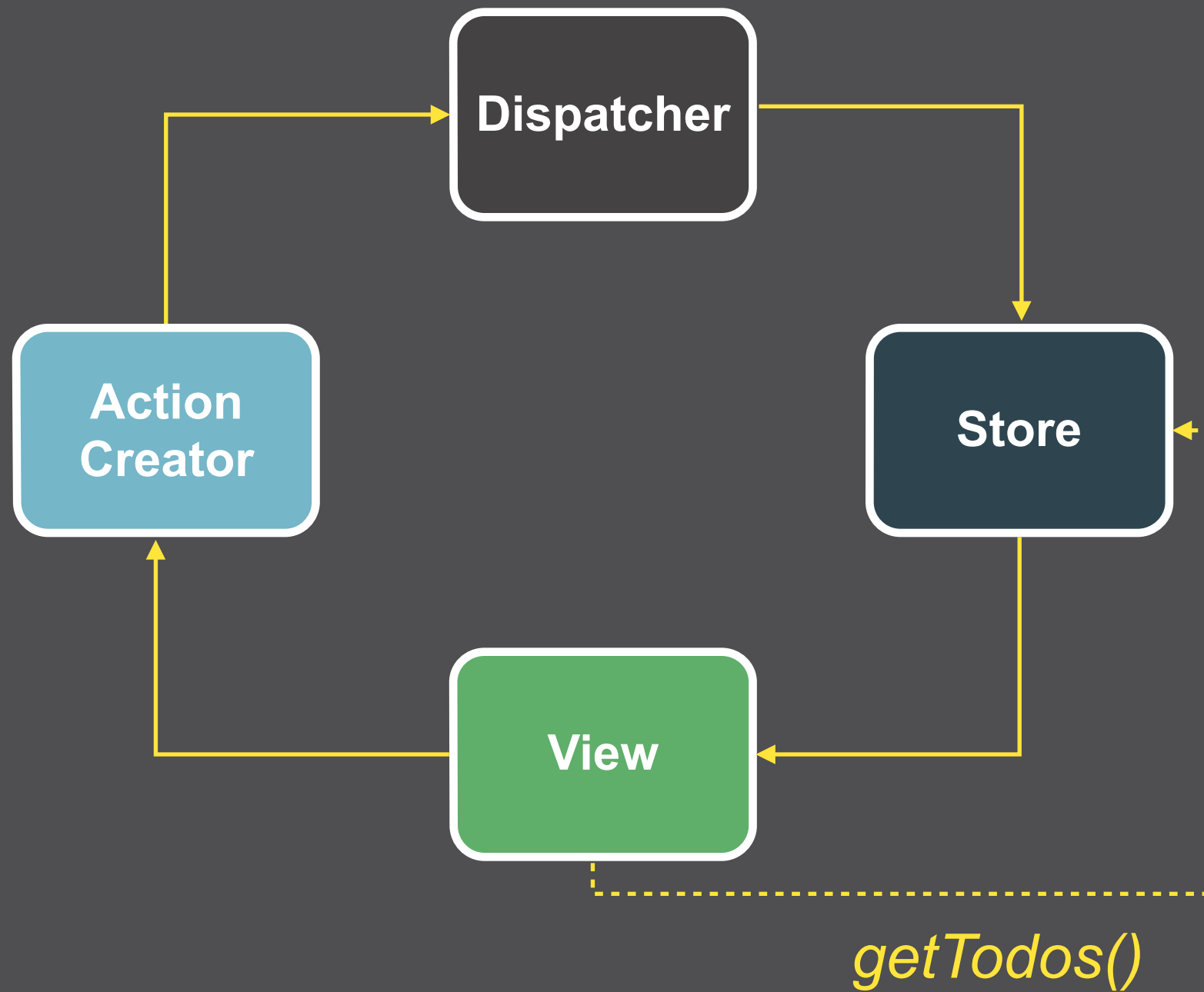


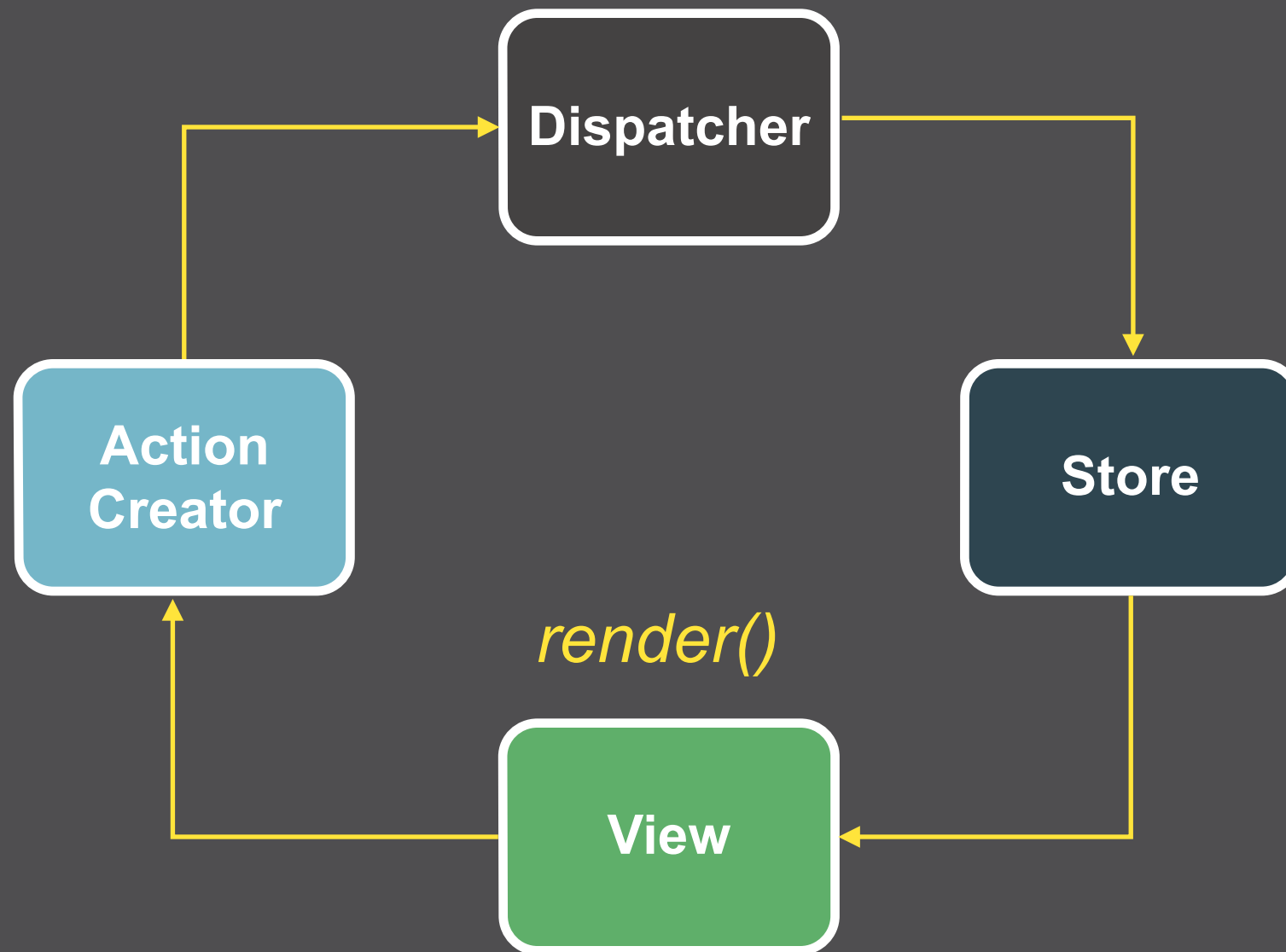


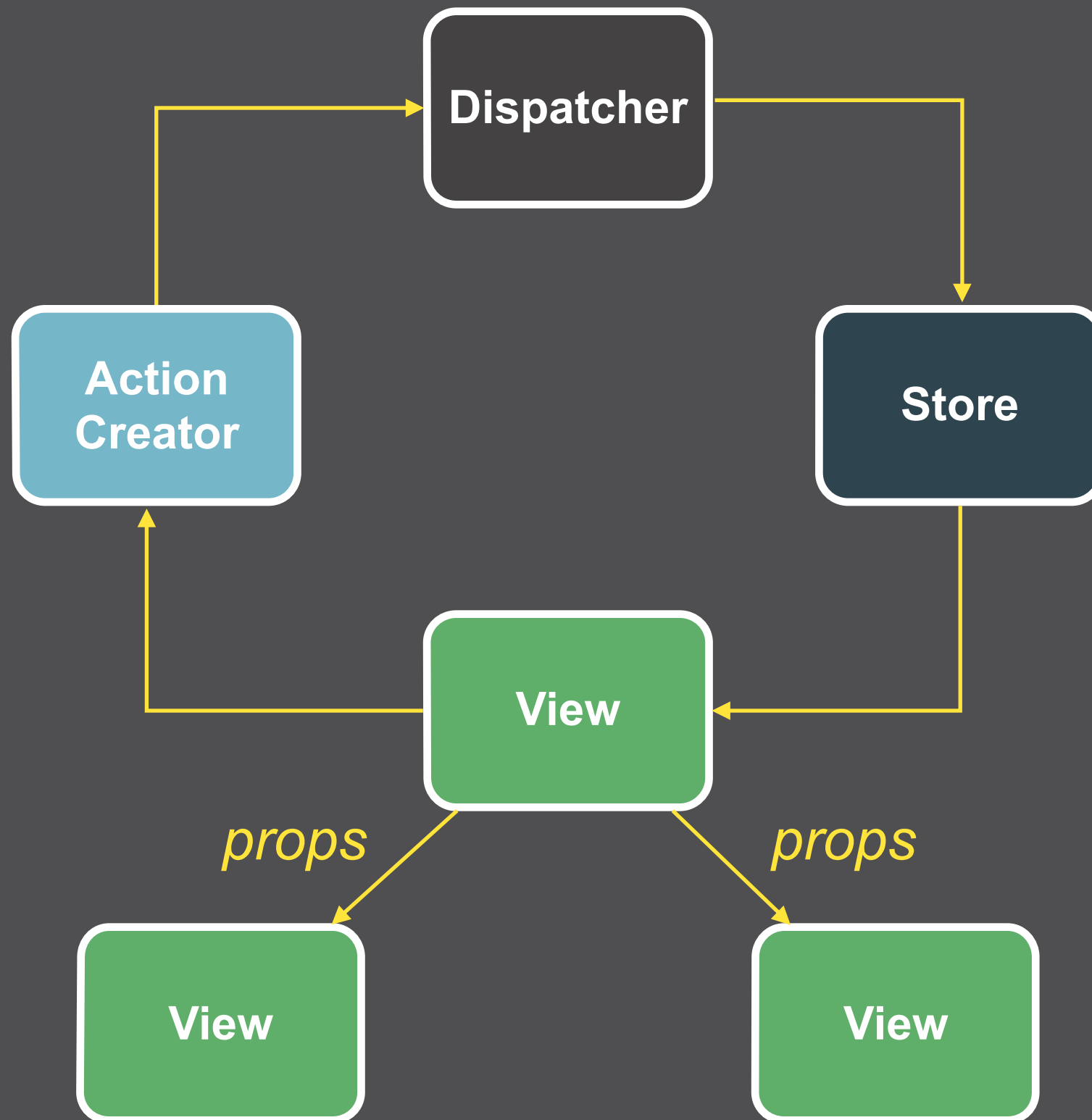


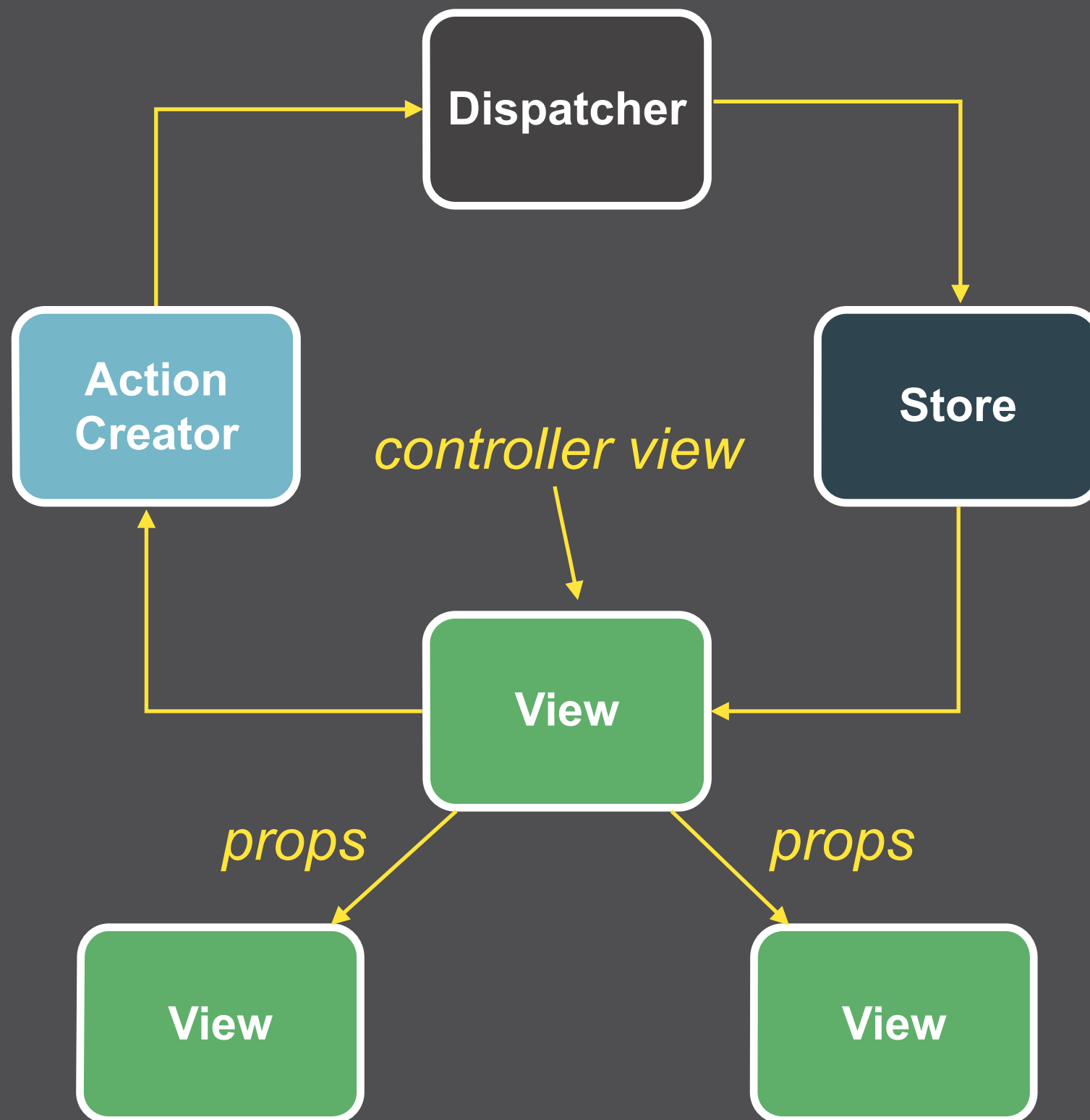




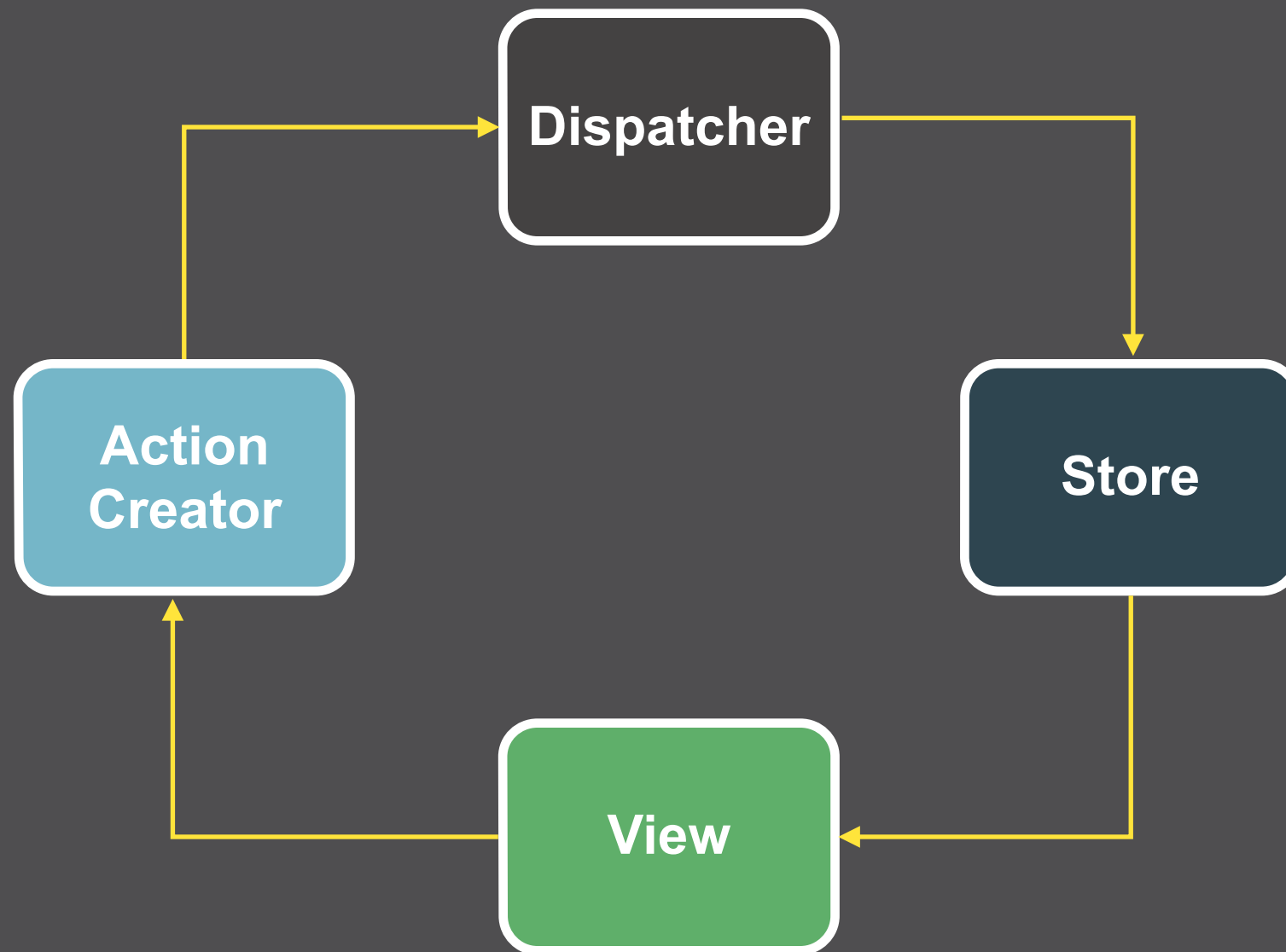


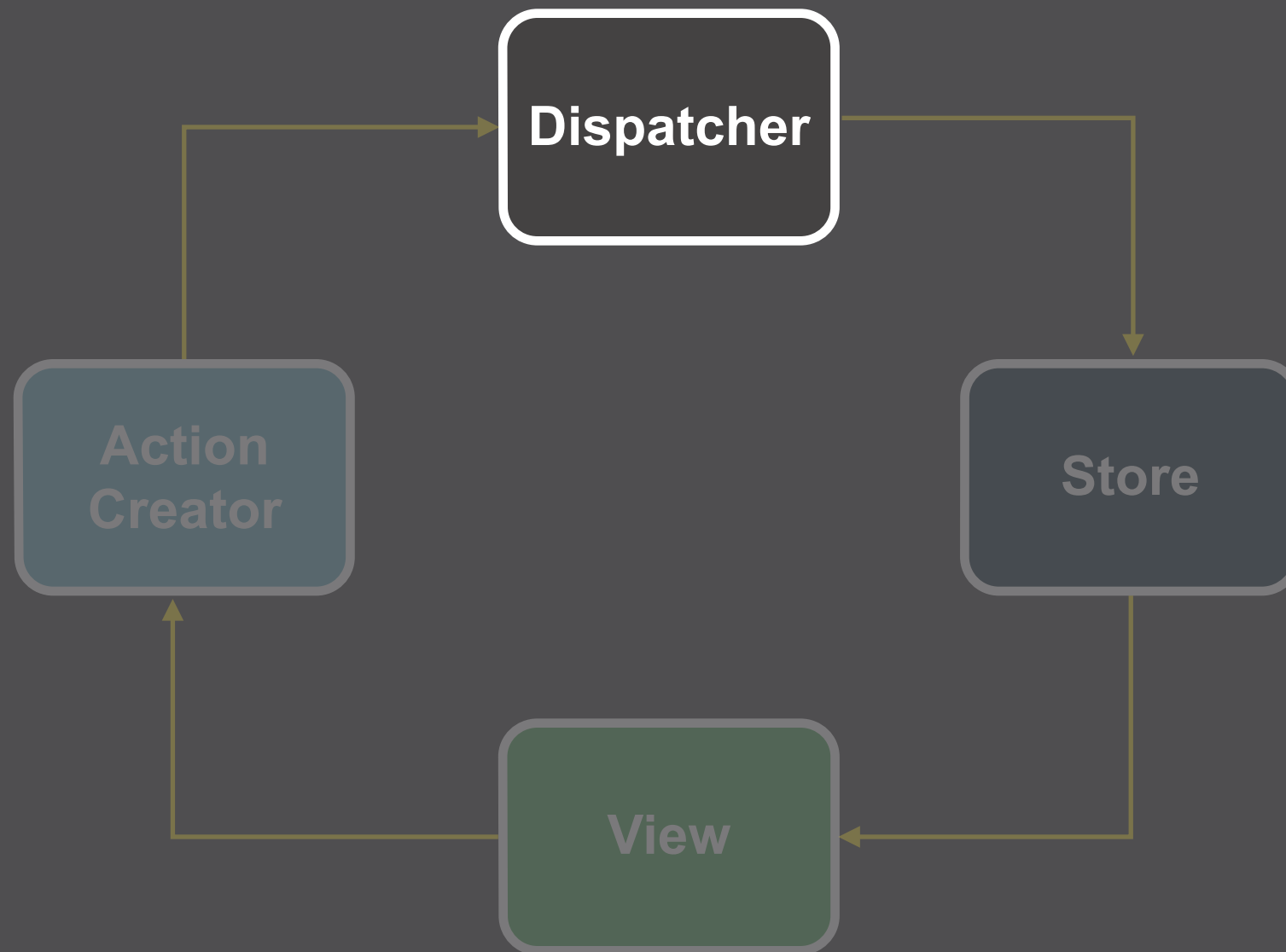




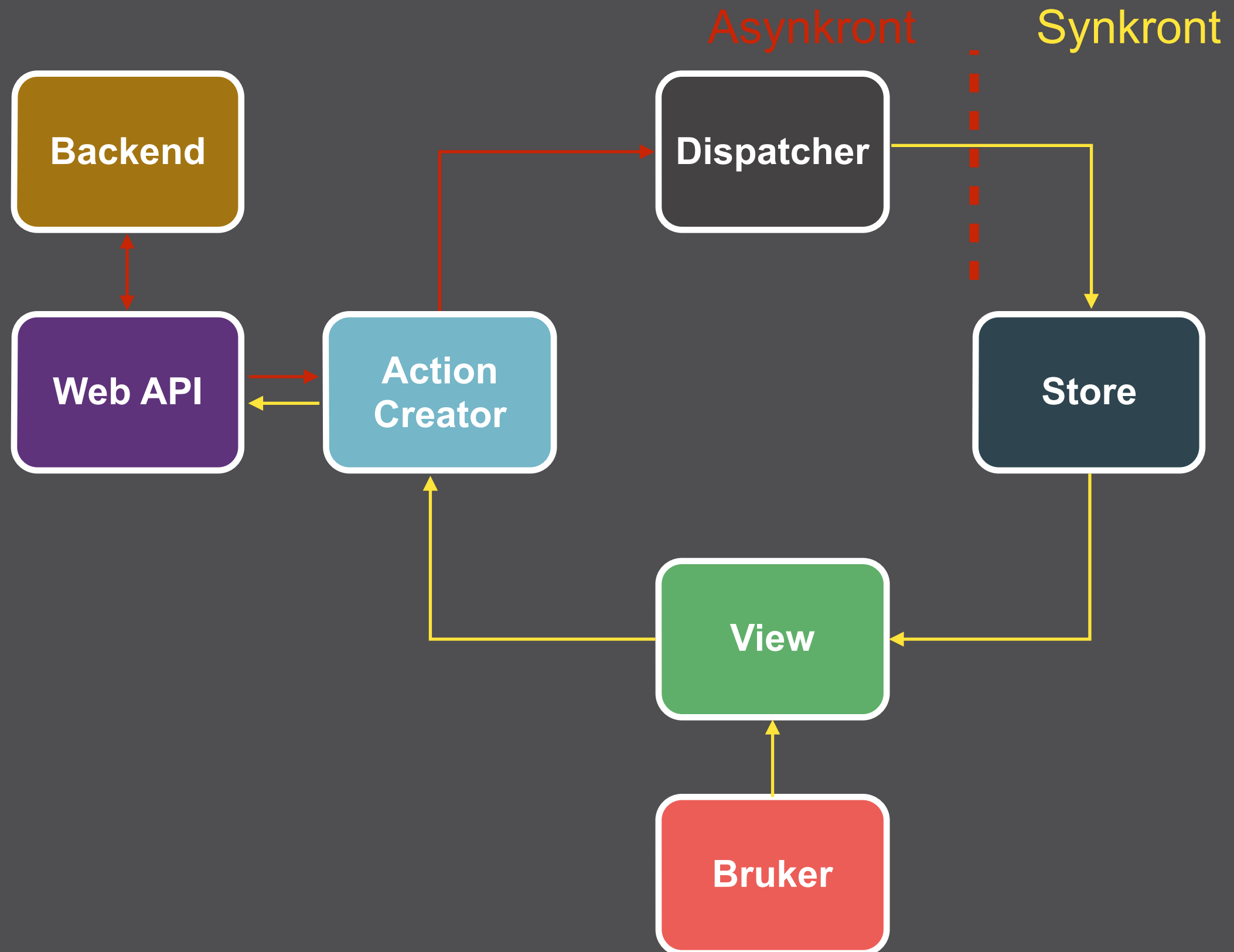


Centrale komponenter

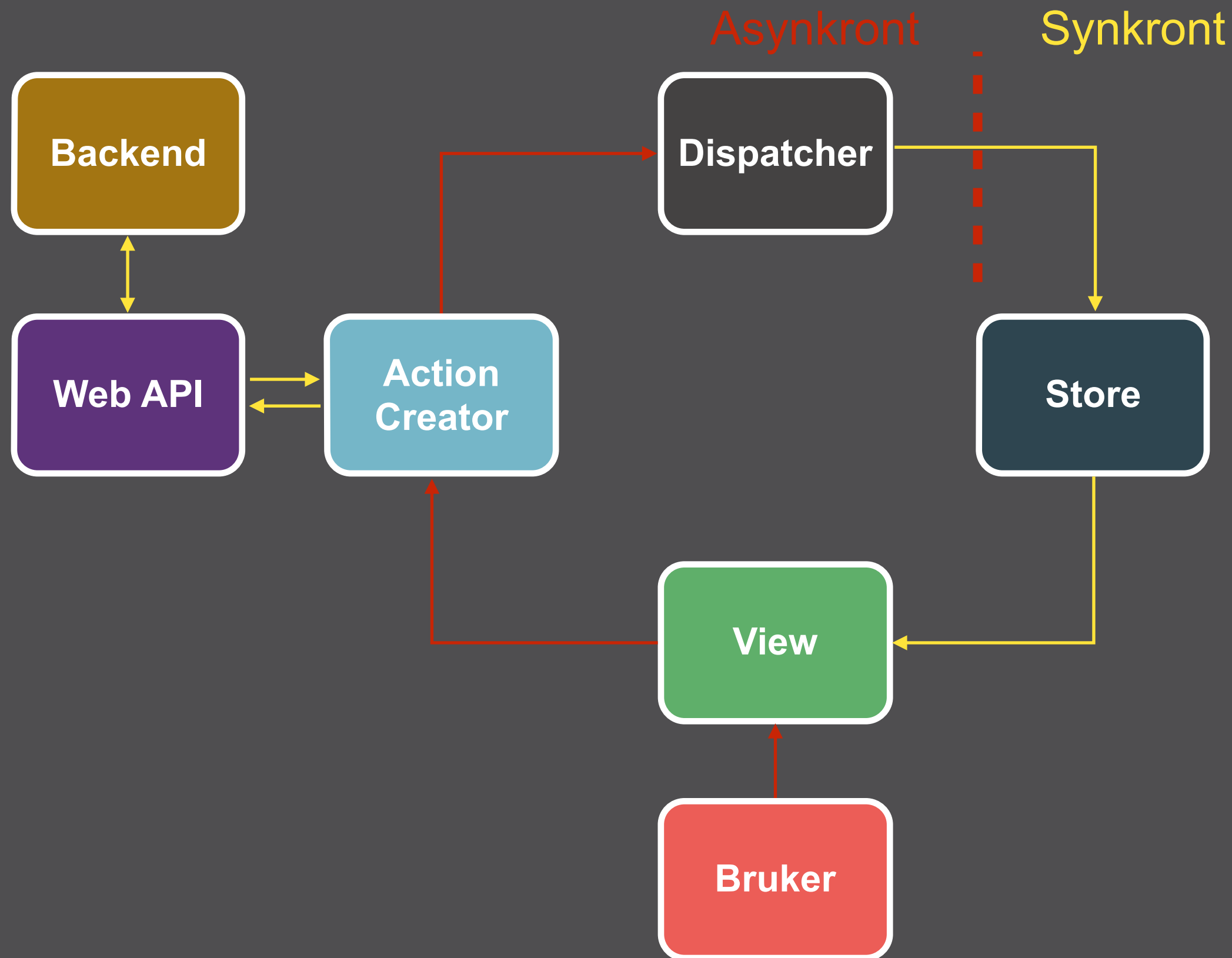




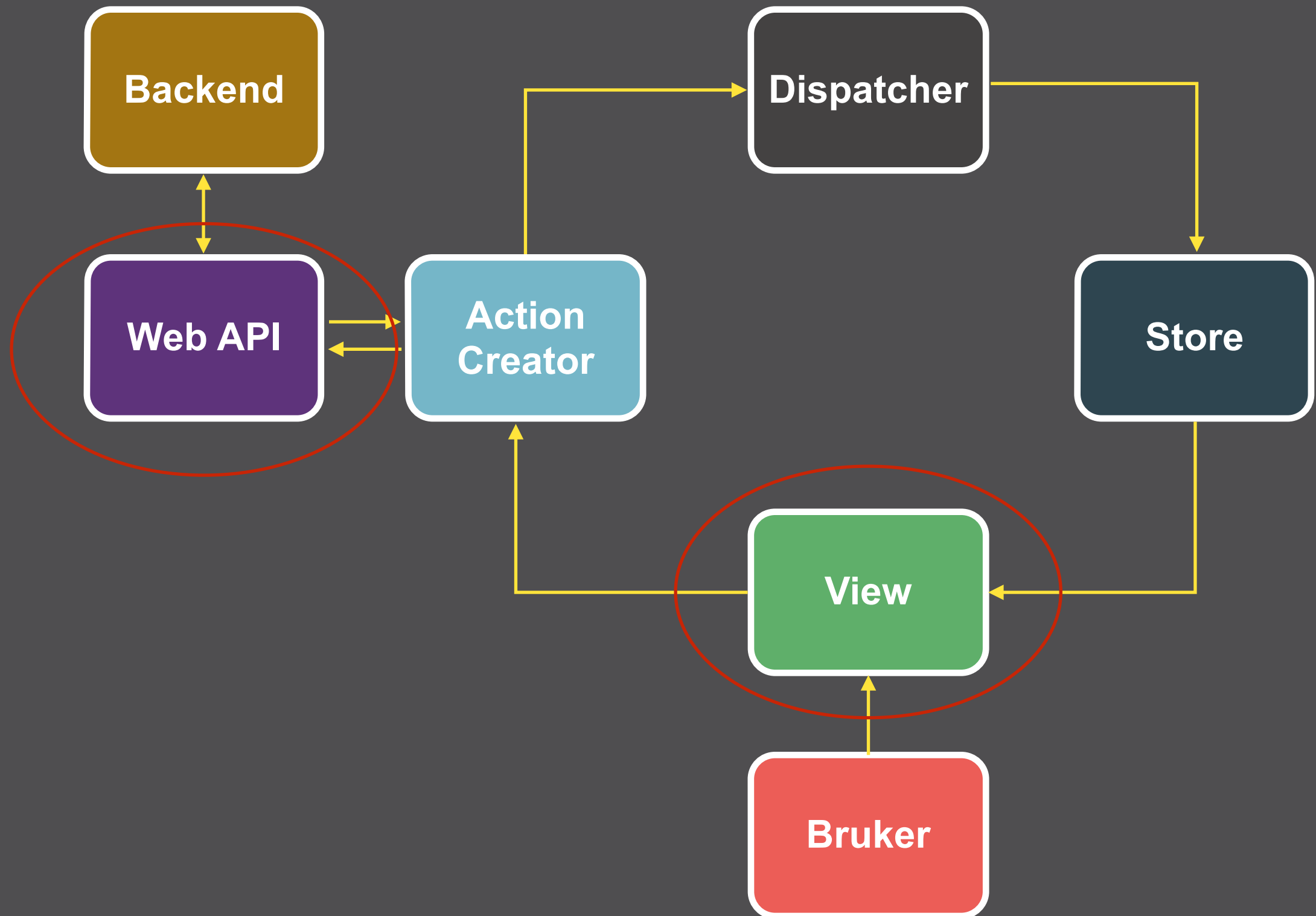
Async “barriere”



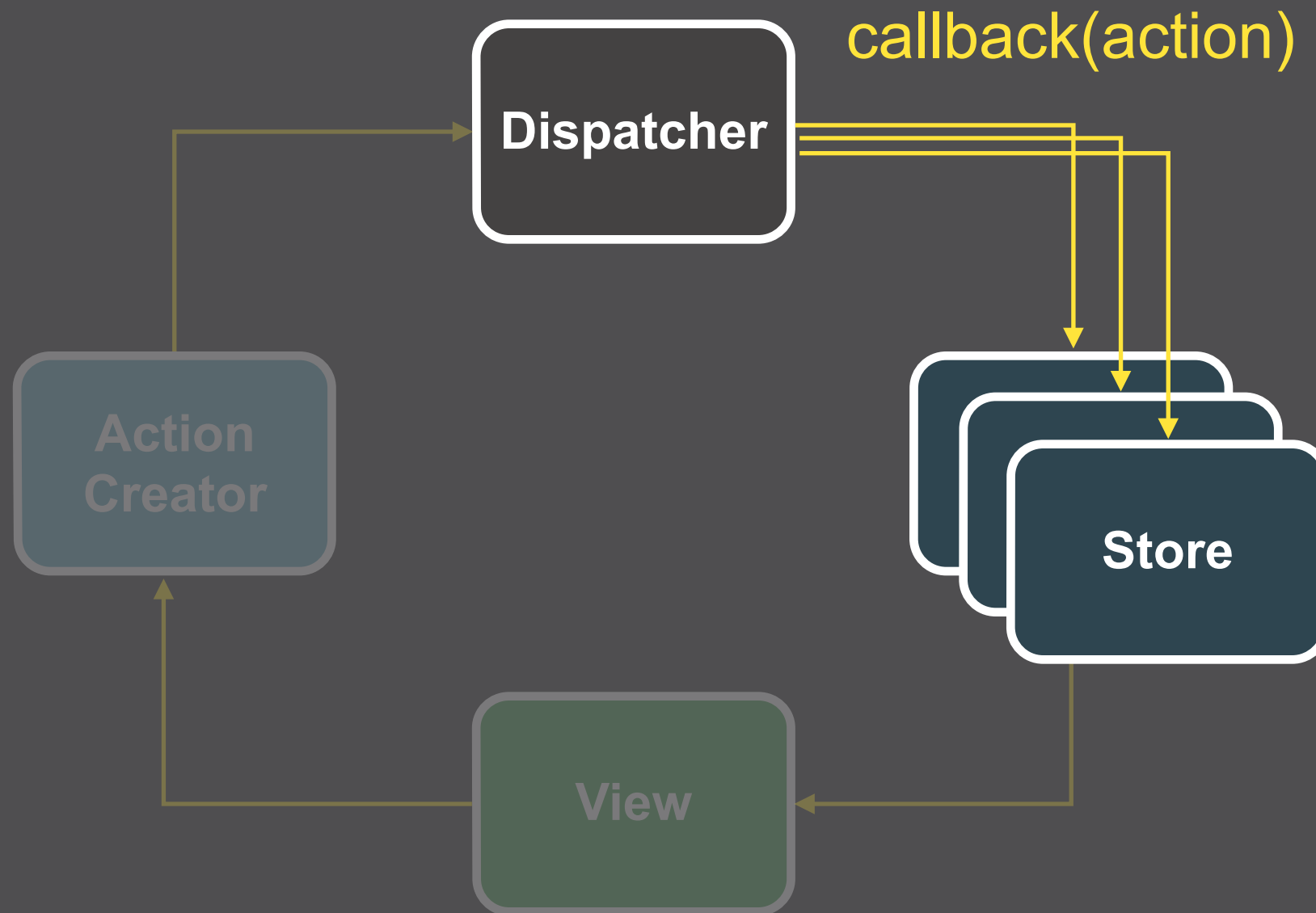
Async “barriere”



Isolering av async



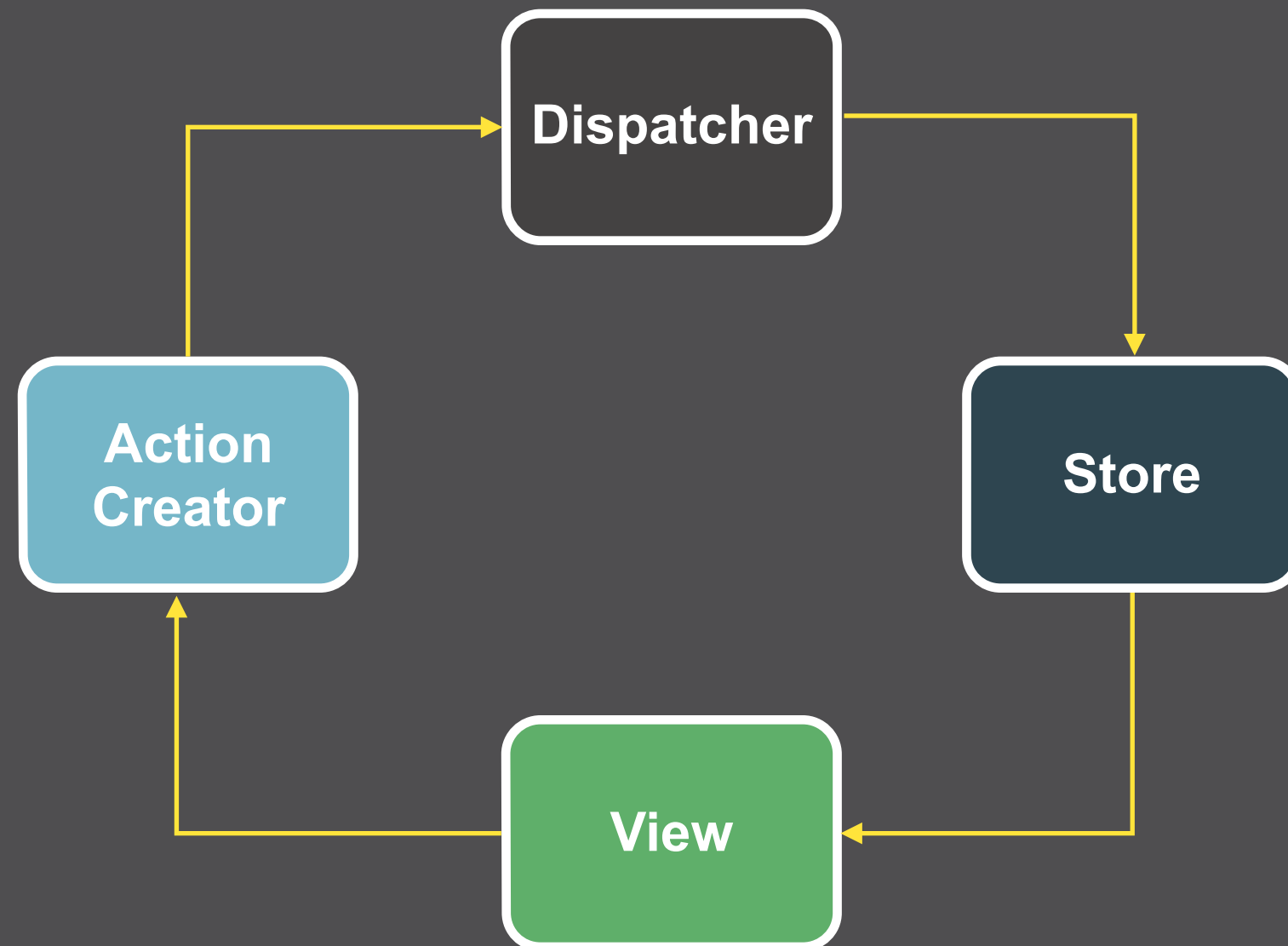
Broadcasting

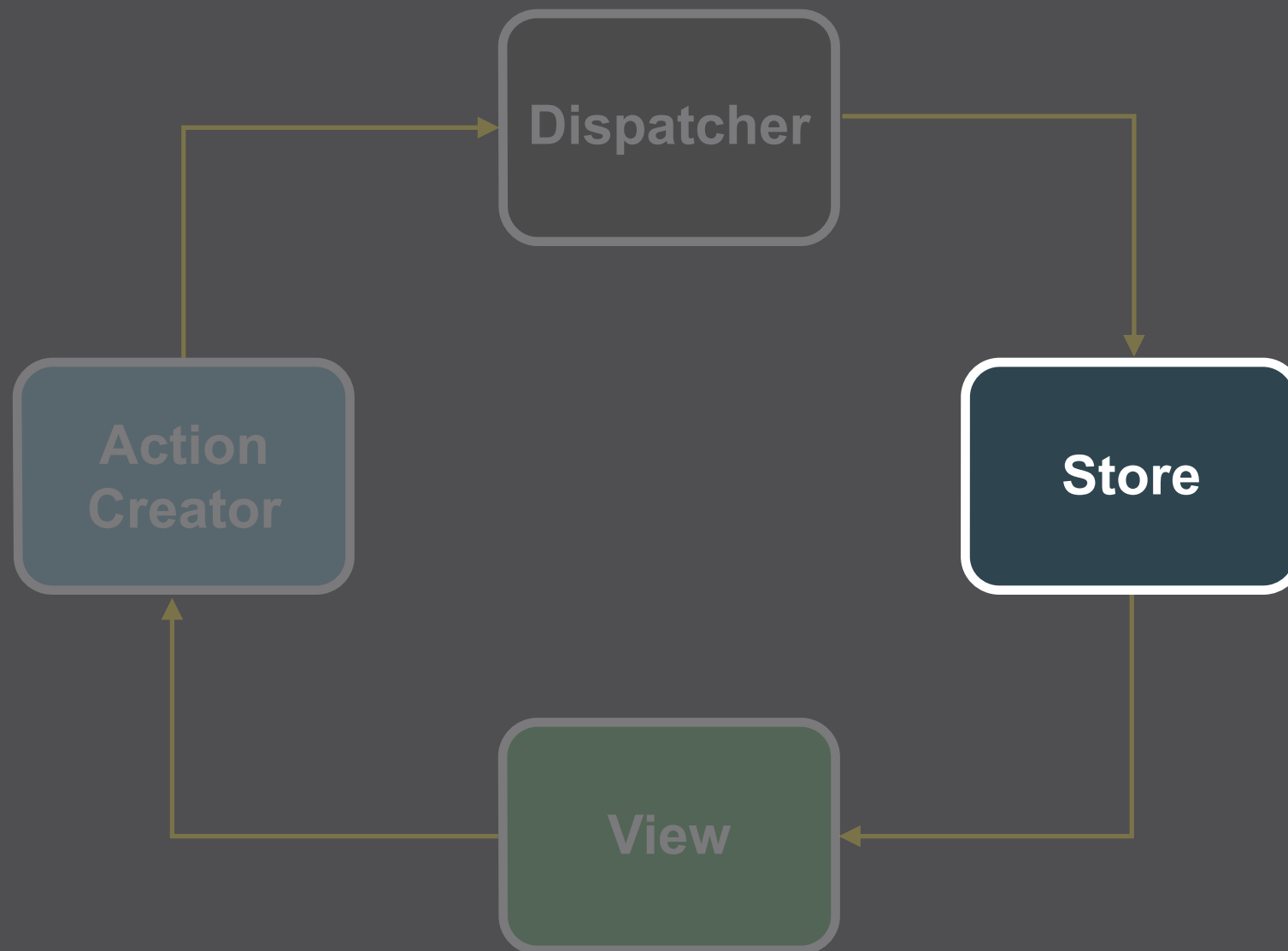




Dispatcher

- Avhengigheter mellom stores: *waitFor([] ids)*
- *dispatch(payload), register(callback), unregister(id)*
- Forhindrer samtidige dispatches
- Barriere mellom asynkront/synkront



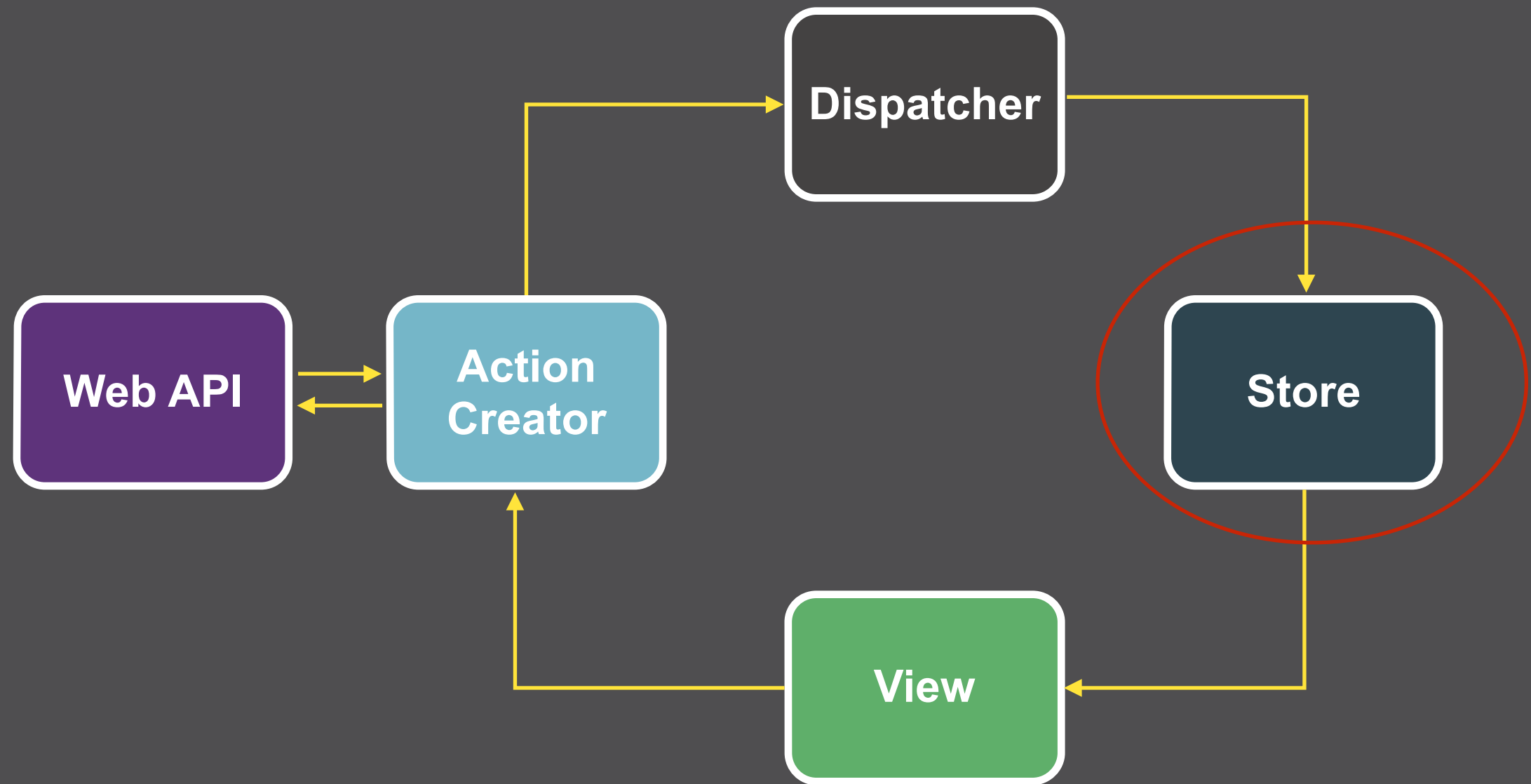


A dark blue rounded rectangle with a white border, containing the word "Store" in white text.

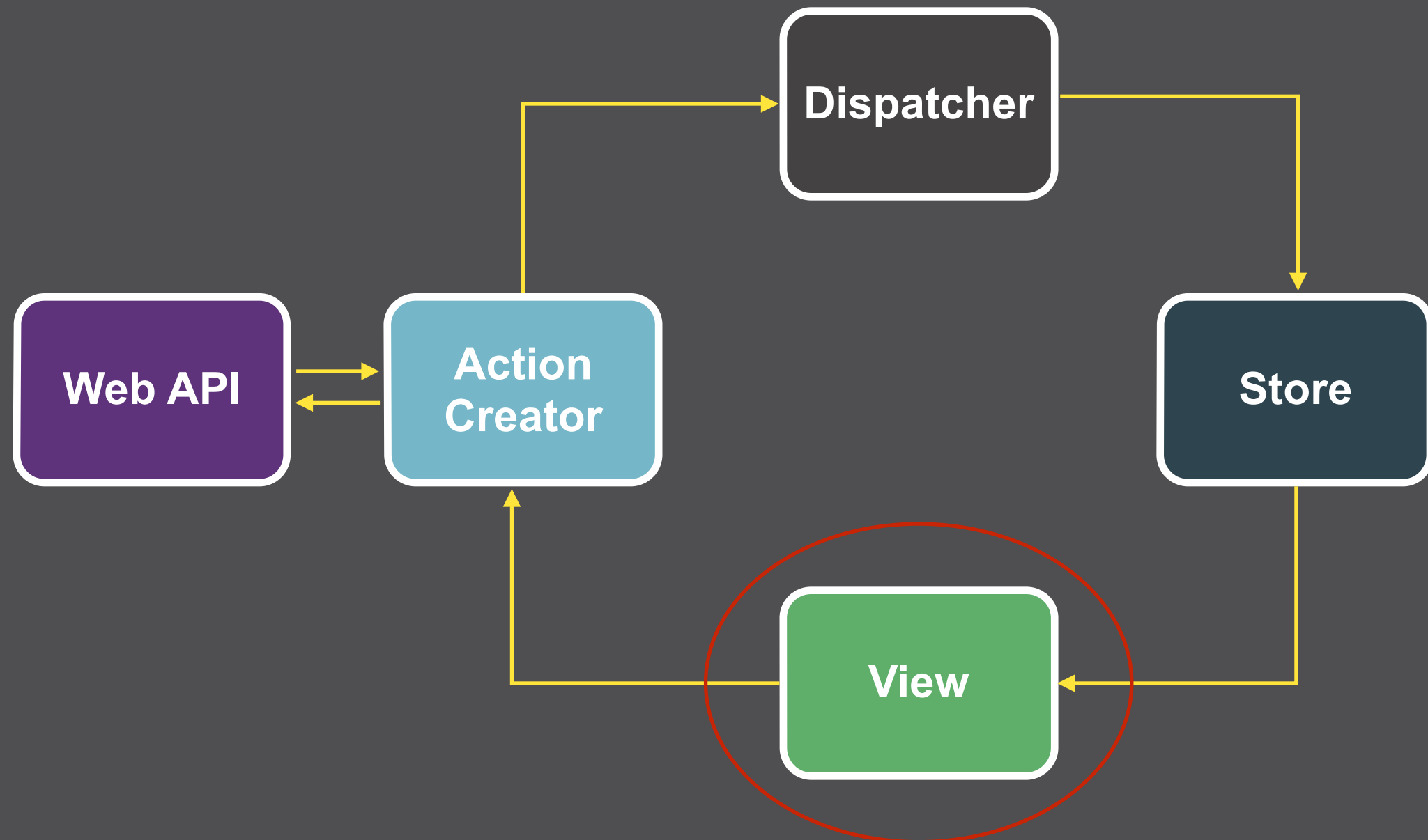
Store

- Registrer seg hos dispatcher vha *register(callback)*
- Inneholder all forretningslogikk og tilstand
- Én store for hvert domene
- Oppdaterer views via events (EventEmitter)

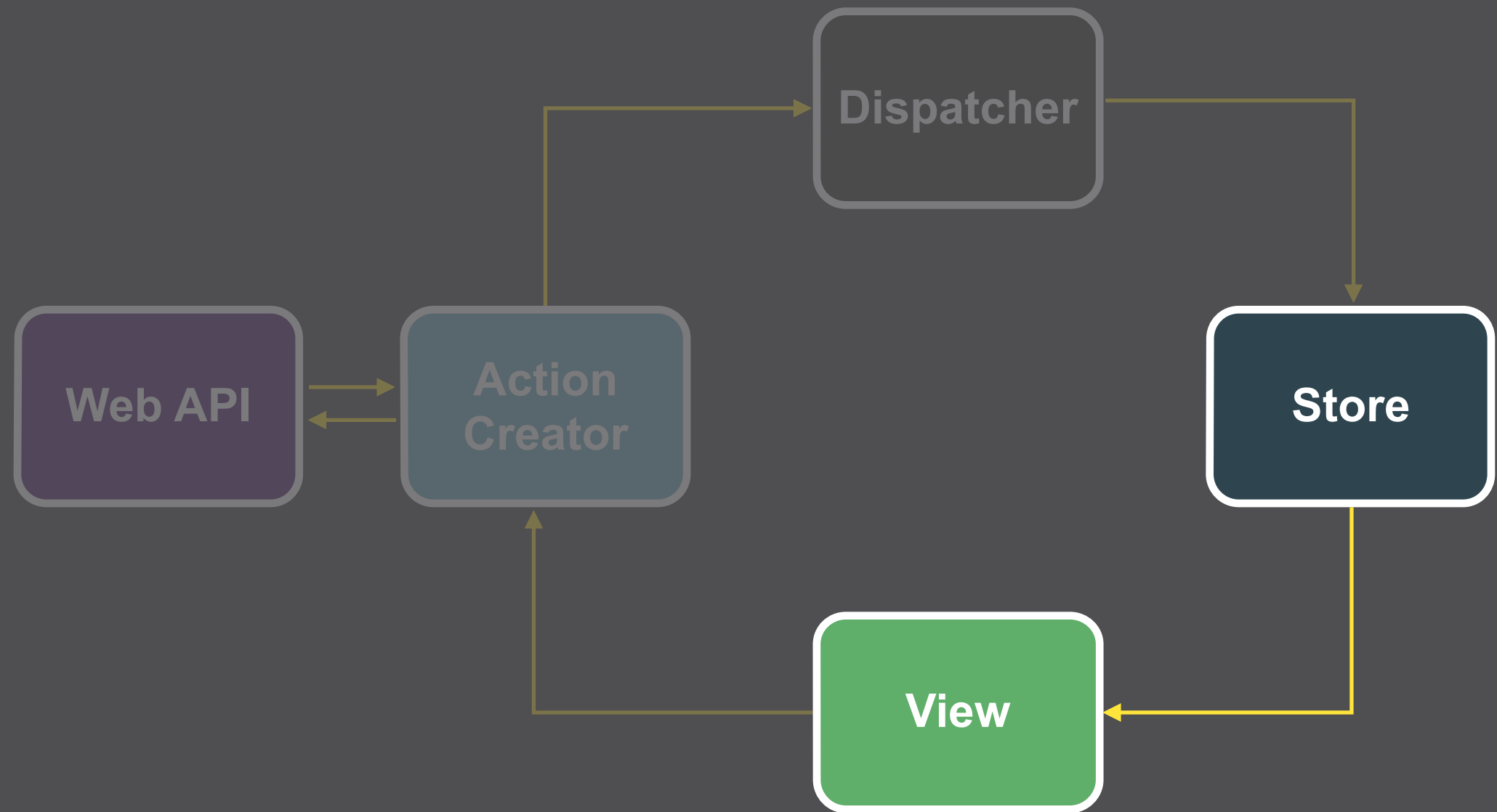
Isolering av tilstand



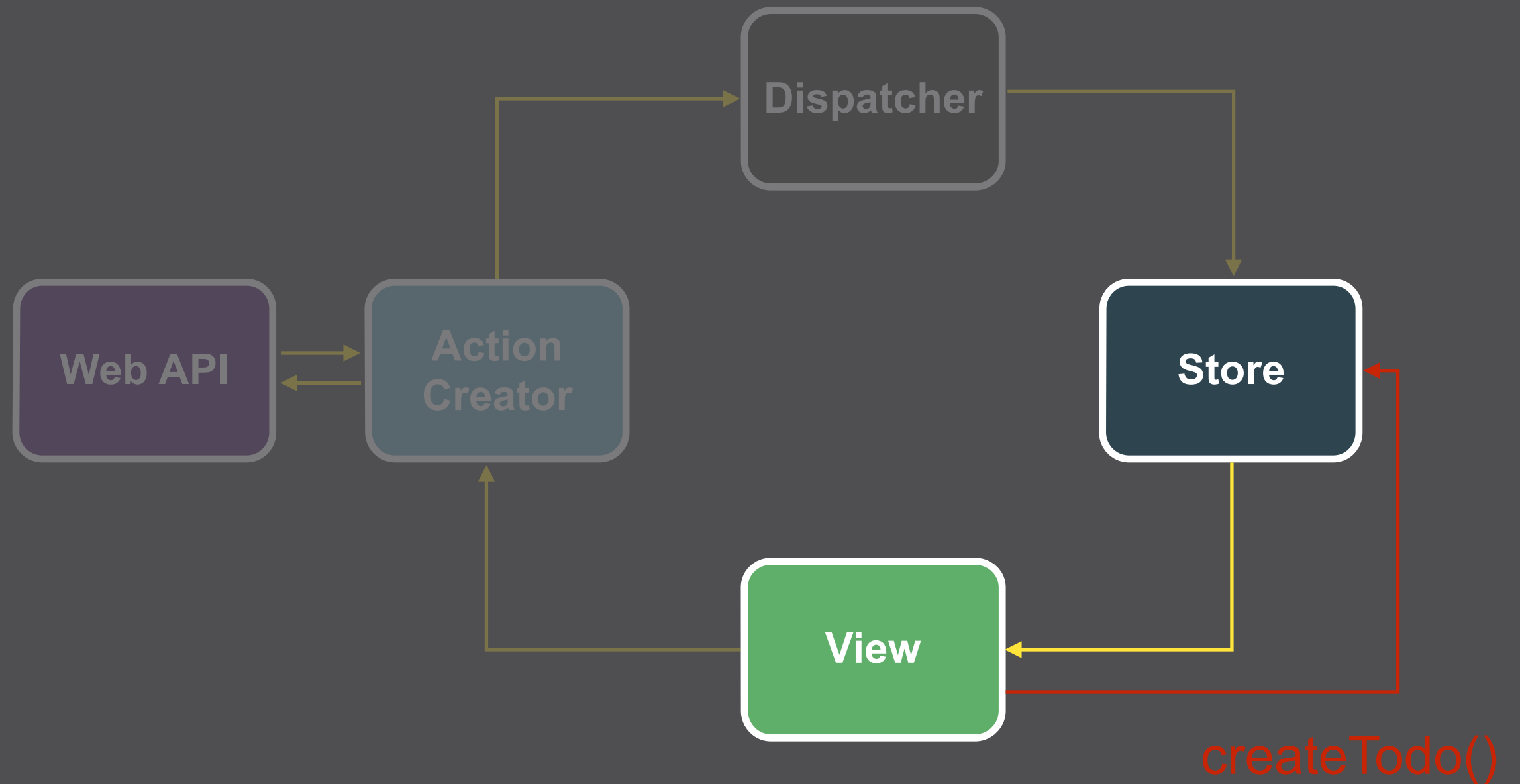
GUI-tilstand i Views



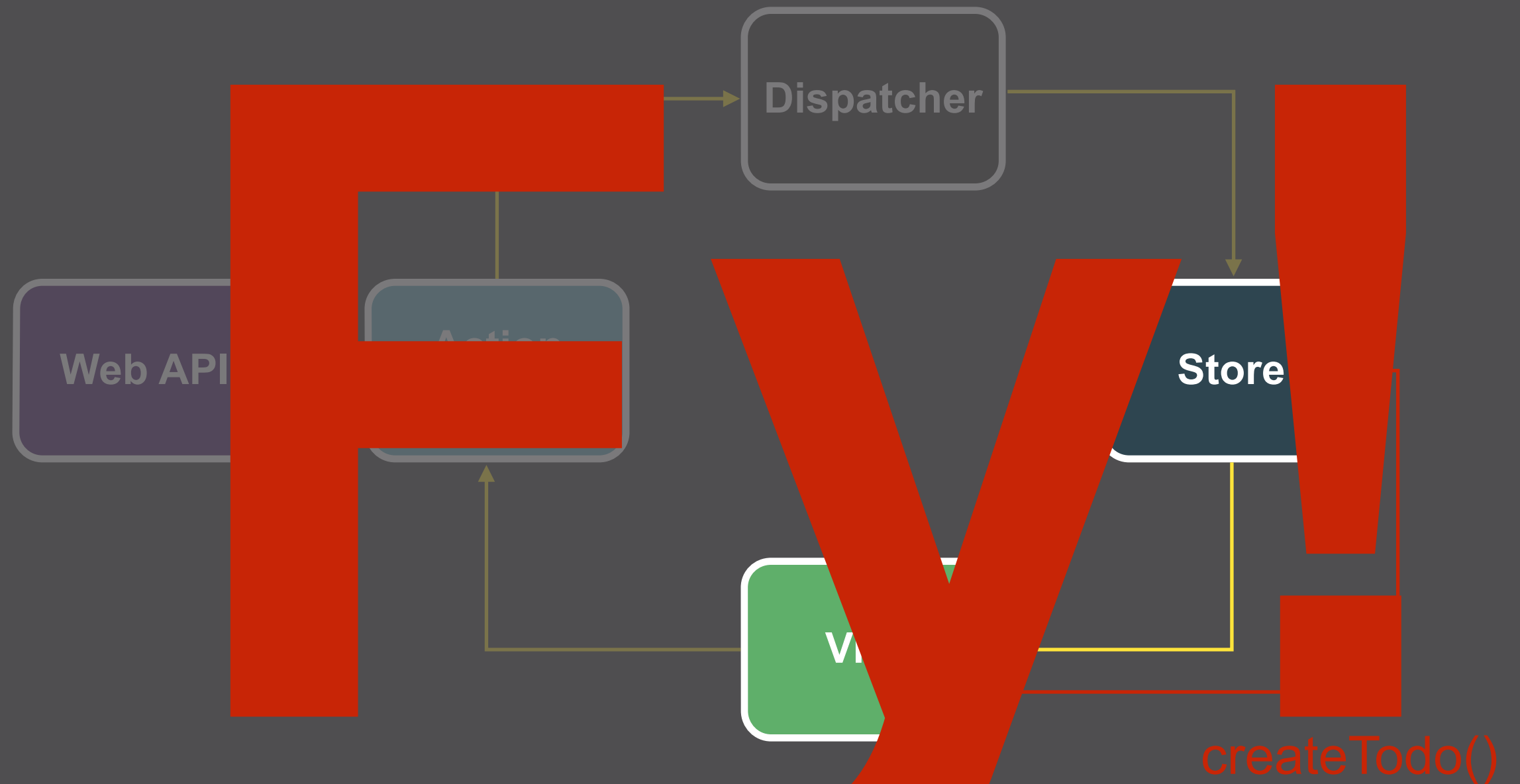
Kun lesetilgang

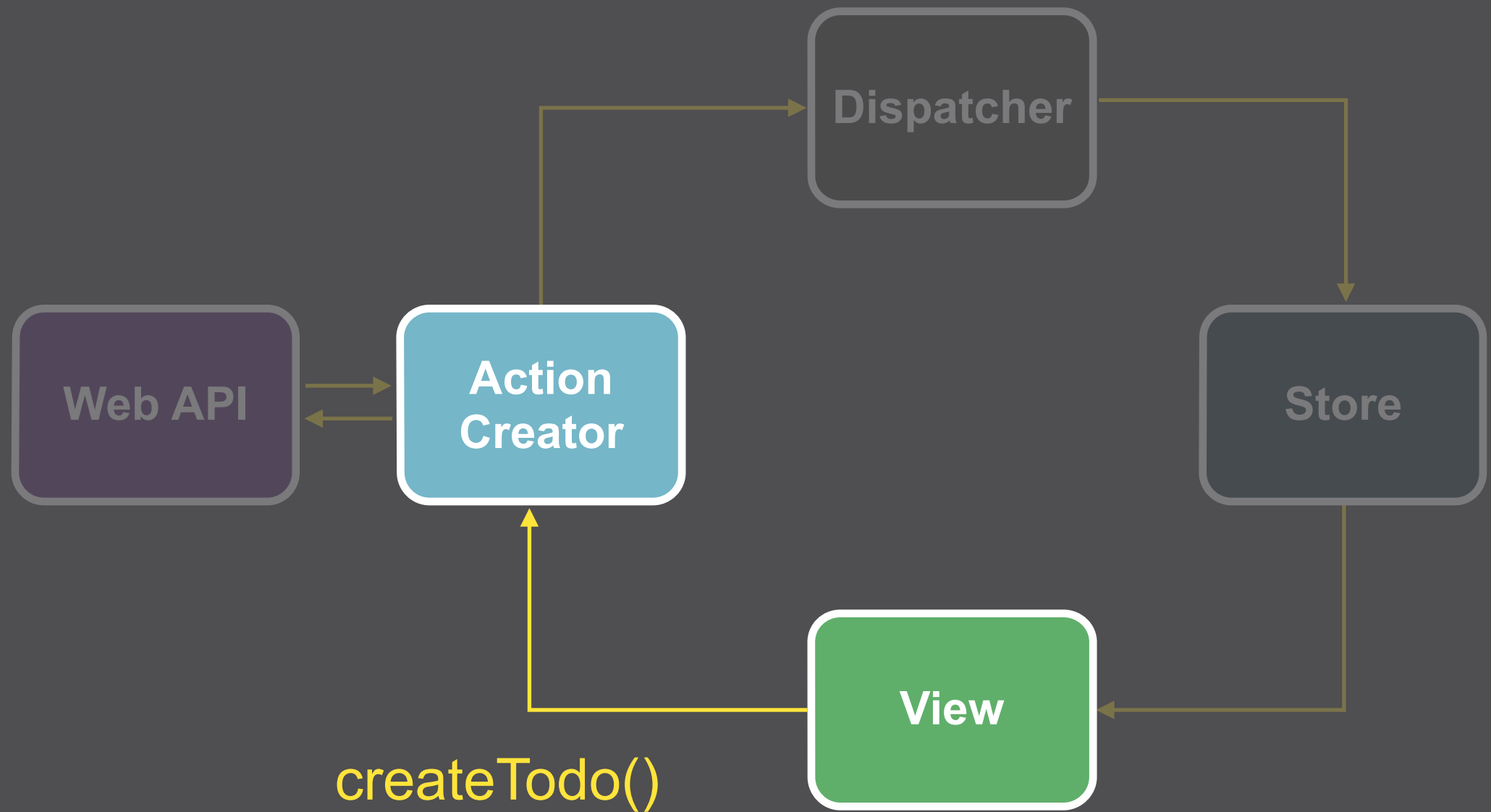


Kun lesetilgang



Kun lesetilgang

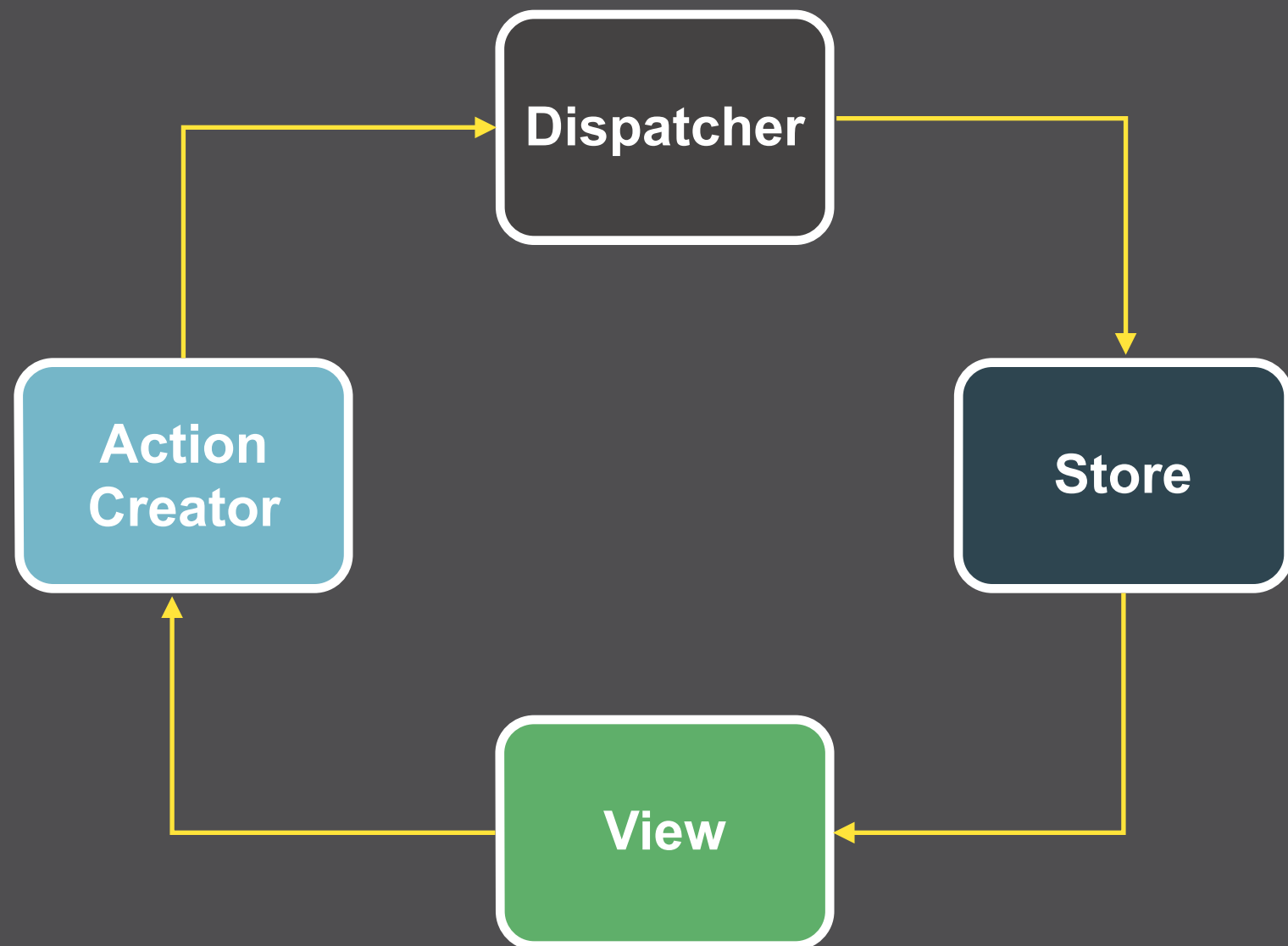




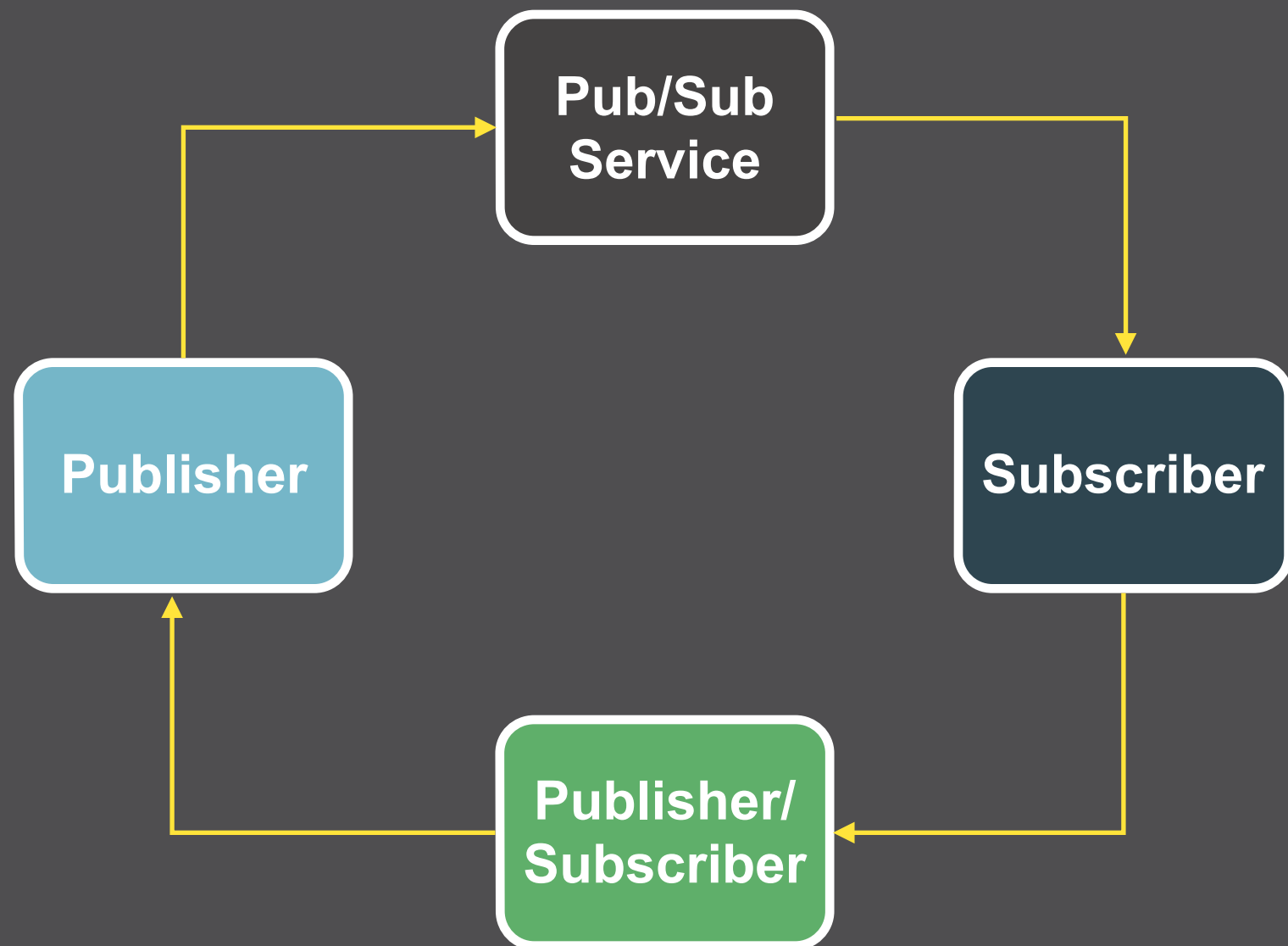


- Flux er ment som et komplement til React
- Viewet på rotnivå er et *controller view*
- Mottar events fra stores
- Kun lesetilgang til Stores

Flux minner meg om...



Pub/Sub



Pub/Sub

“Dispatcher is used to broadcast payloads to registered callbacks. This is different from generic pub-sub systems in two ways:

- 1. Callbacks are not subscribed to particular events.*
- 2. Every payload is dispatched to every registered callback.*

Callbacks can be deferred in whole or part until other callbacks have been executed.” [0]

[0]: <http://facebook.github.io/flux/docs/dispatcher.html>

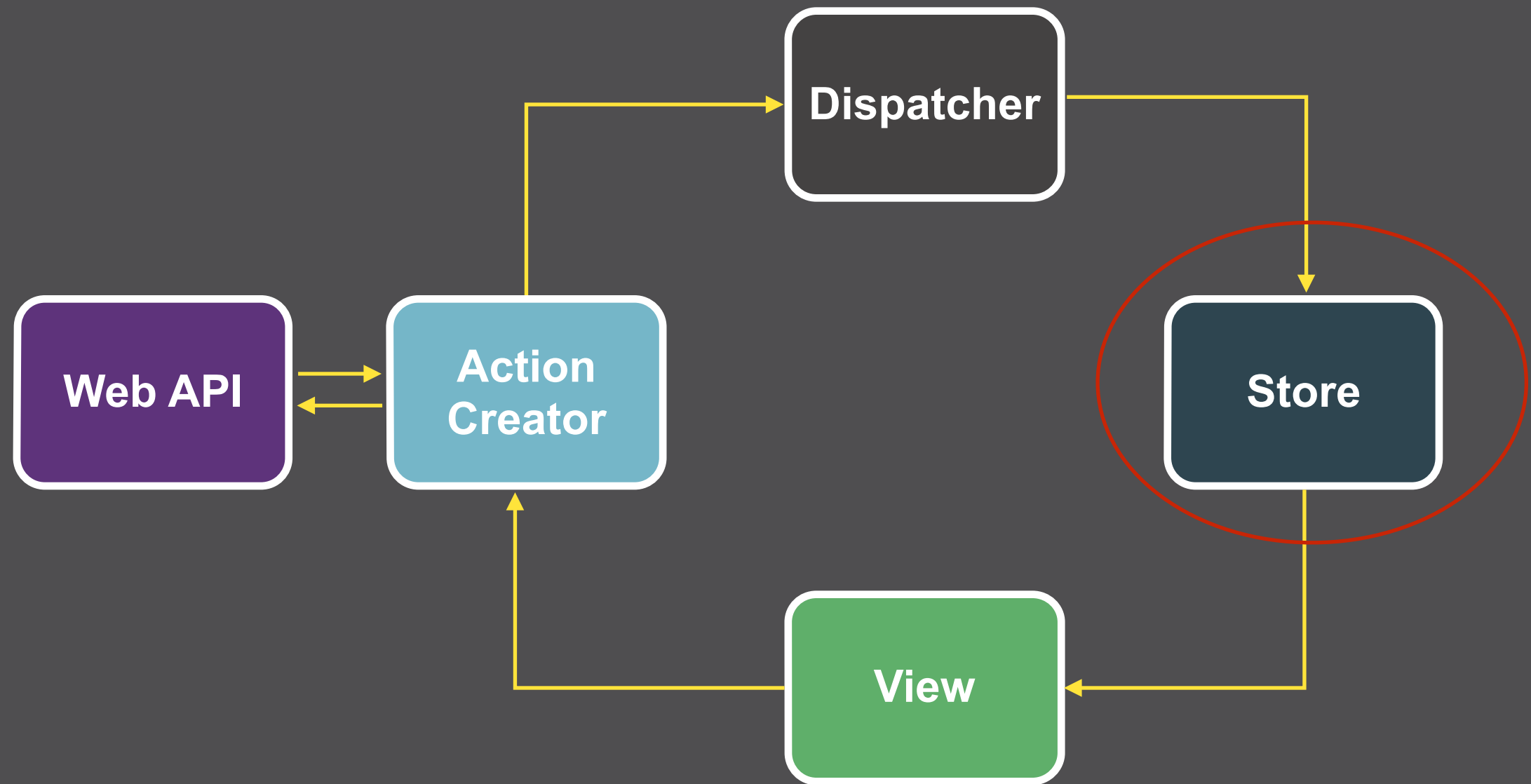
Hvorfor?

Hvorfor Flux?

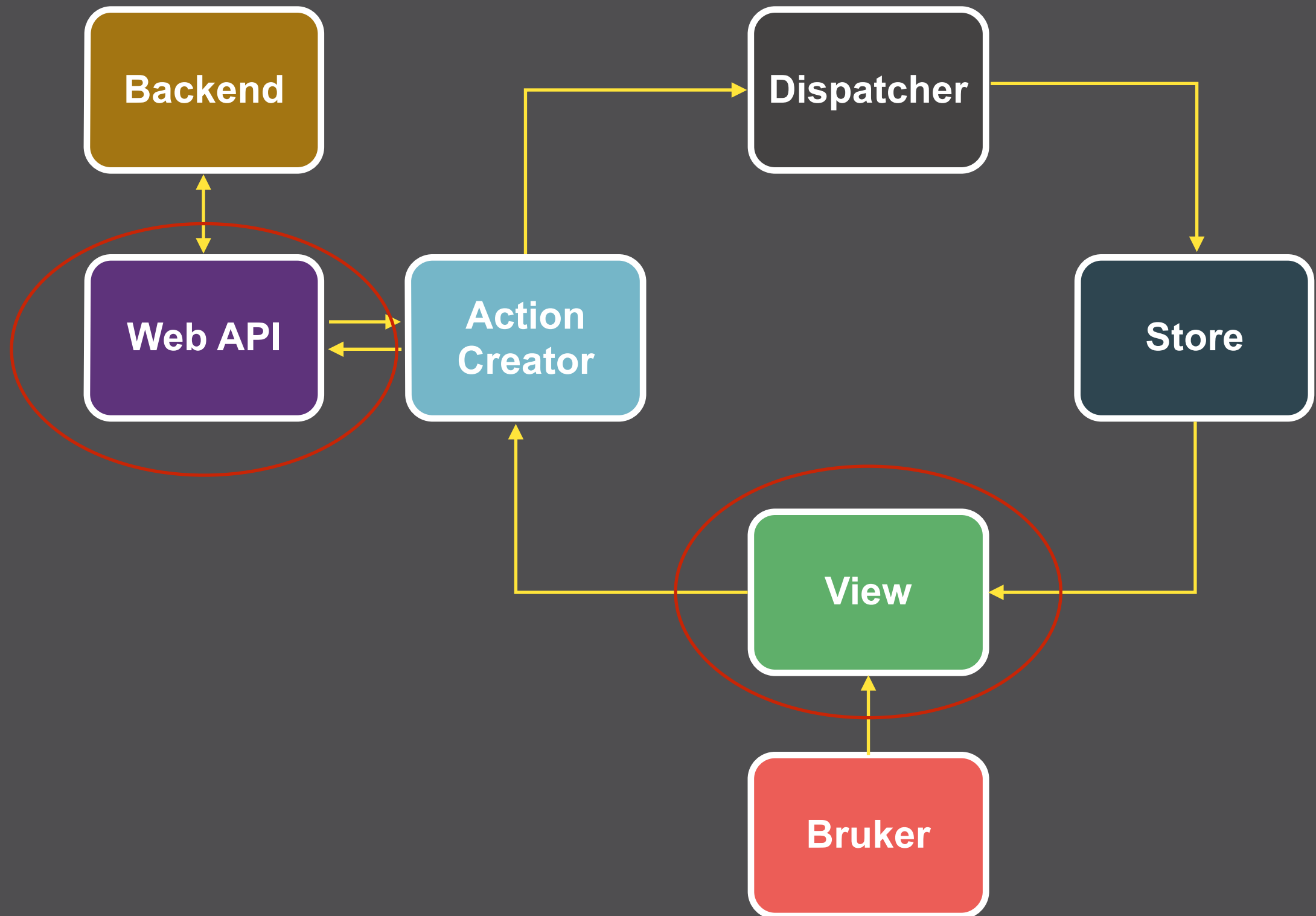
- Konseptuelt enkelt
- Forutsigbart, lett å debugge
- Lettere å resonnerer rundt
- Isolerer asynkronitet og tilstand

Oppsummering

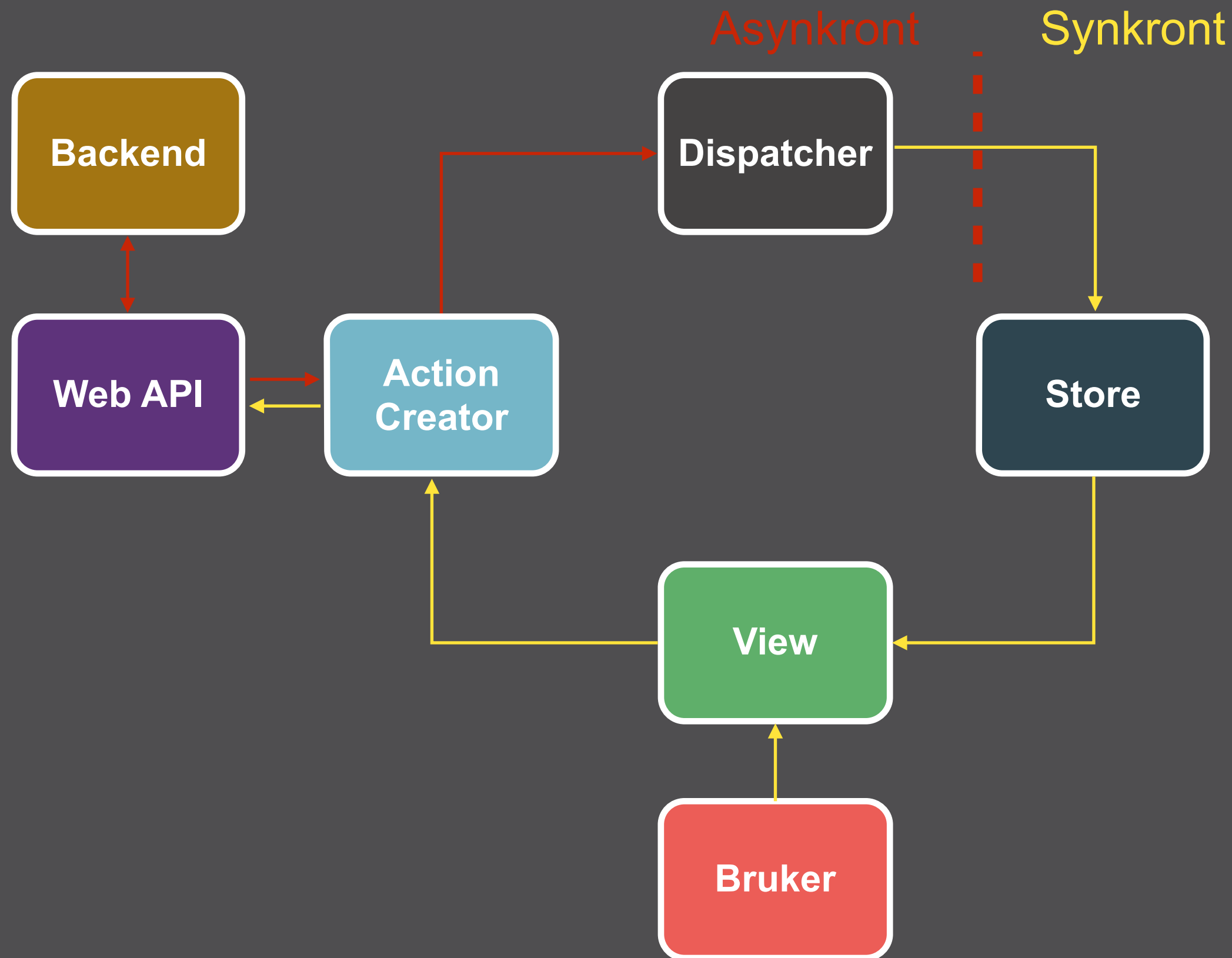
Isolering av tilstand



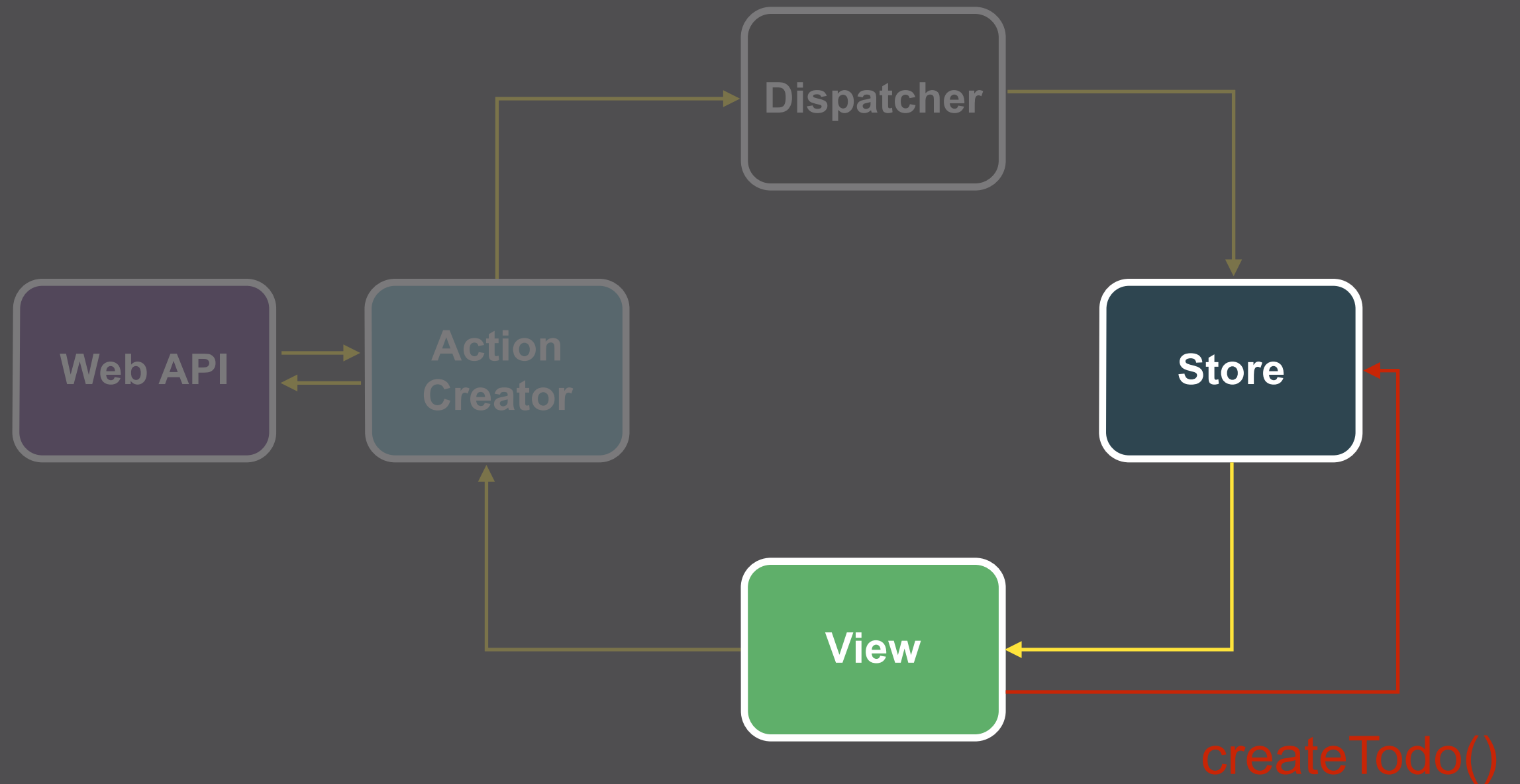
Isolering av async



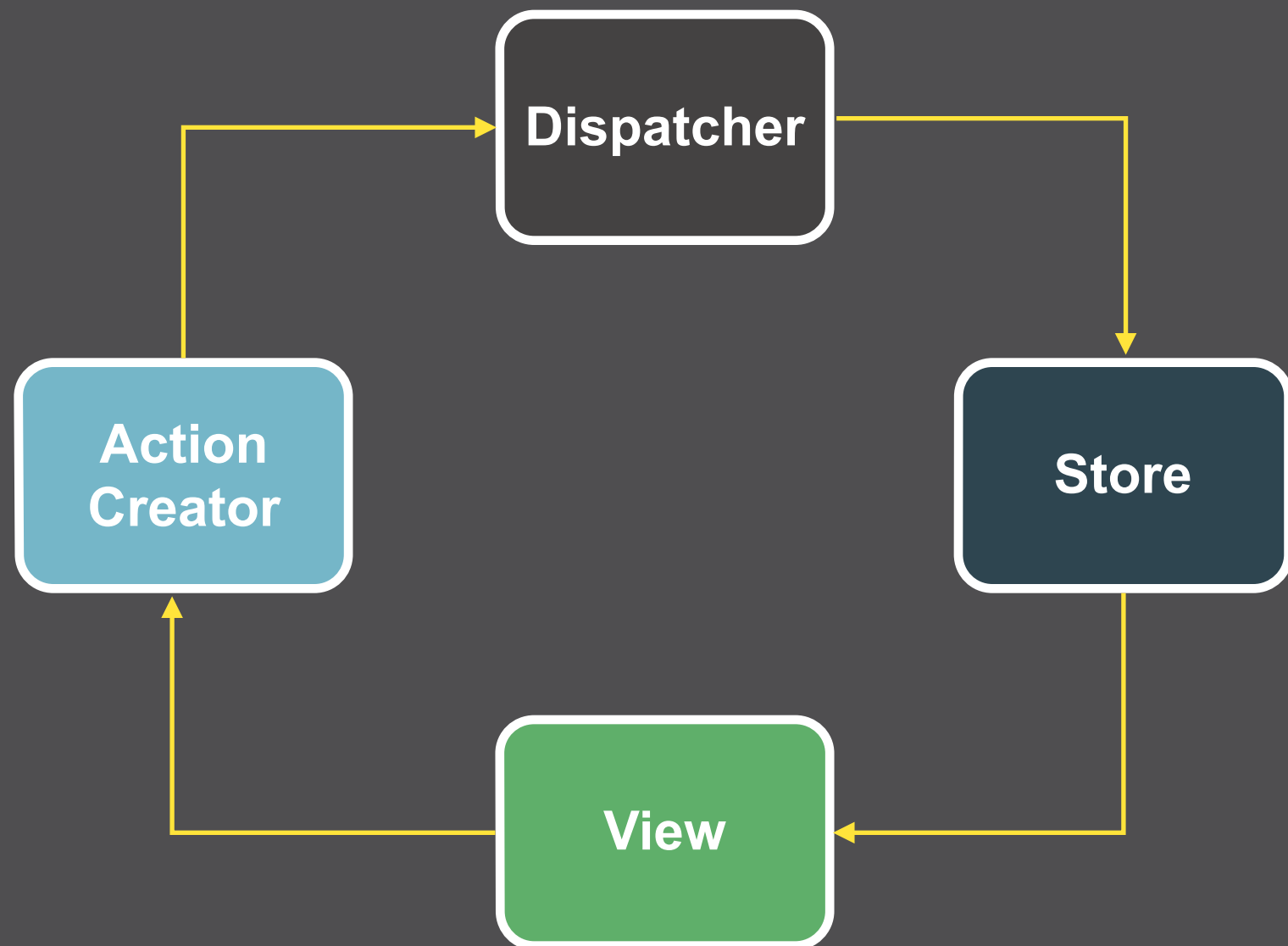
Async “barriere”



Kun lesetilgang



Mental modell



Framover

- Finsliping av konseptene (reflux, fluxxor, fluxible)
- Mer påvirkning fra funksjonelle paradigmer (frp)
- om (Clojurescript), omniscient (JS)
- Flow, Typescript

<https://github.com/kjbekkelund/flux-workshop>